Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Unidade Acadêmica de Engenharia Elétrica
Sistemas de Automação Industrial

Controladores Lógicos Programáveis

Bolsista: Vanderley Maia Gomes
Orientador: Péricles Rezende Barros
Programa: Grupo PET de Engenharia Elétrica
Tutor PET: Edmar Candeia Gurjão

Sumário

Capítulo 1

Conceitos Gerais

1.1 Introdução

Algumas tarefas devem ser realizadas inúmeras vezes de forma igual e com certa periodicidade, nesse tipo de trabalho a execução humana deixa muito a desejar, visto que o ser humano facilmente se cansa de atividades repetitivas e com o tempo sua eficácia tende a cair muito.

Essas tarefas podem ser realizadas por máquinas que possam ser programadas pelo ser humano. O dispositivo a ser programado é o Controlador e ele é responsável por todas as ações do restante do sistema.

O controlador estudado nesse curso é o CLP, controlador lógico programável que será explicado em maiores detalhes.

Um exemplo de um processo que é repetitivo e necessita ser controlado é o engarrafamento de líquidos, tais como refrigerantes, este processo pode ser visualizado na figura 1.1. Uma esteira movimentada por um motor ligado a saída 3 do CLP, deve levar os vasilhames vazios através de um processo por onde estes serão preenchidos por refrigerante. Para detectar se o vasilhame está na posição correta para receber o refrigerante, um sensor (S1) é colocado na esteira, esse sensor é ligado a Entrada 1 do CLP a qual é lida periodicamente pelo programa em execução, quando o vasilhame chegar a posição do S1 a leitura irá acusar que o mesmo está no local correto e então a saída 1 do CLP é ativada ligando assim o atuador 1 (A1) que pode ser implementado por uma válvula solenóide. Enquanto o vasilhame é preenchido a esteira fica parada, portanto é necessário conhecer o tempo que leva para o vasilhame ser cheio a fim de, passado esse tempo o Atuador 1 é desligado e o Atuador 3 (Motor da Esteira) é novamente ligado. Após o vasilhame ser cheio a esteira volta a se movimentar levando-o até o sensor 2 (S2) que está ligado a entrada 2 do CLP, a qual, semelhante a entrada 1, é lida periodicamente,

quando essa leitura acusar que o vasilhame se encontra no local do sensor então o atuador 2 (A2) é ativado pelo CLP através da saída 2, será então colocada uma tampa finalizando o processo de engarrafamento. A esteira volta a se movimentar e os vasilhames podem ser coletados por uma pessoa posicionada ao fim da esteira.

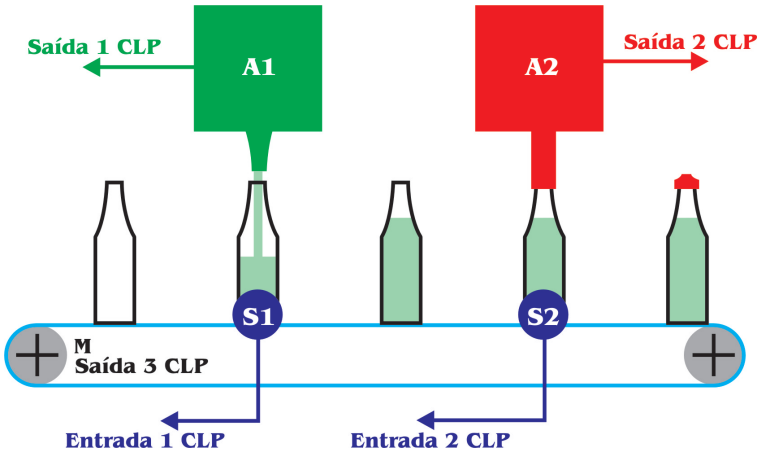

Figura 1.1: Processo de Engarrafamento

1.2 Sistemas de Controle Microprocessados

Quando temos um sistema microprocessado que controla um processo qualquer, podemos sem muita dificuldade utilizar esse mesmo sistema para controlar uma grande variedade de processos, para isso devemos apenas mudar o programa que será executado no microprocessador. Isso representa uma grande vantagem em relação a sistemas baseados apenas em Hardware, nos quais torna-se muito complexa a tarefa de modificar o processo, uma vez que qualquer mudança na lógica do processo resulta em mudar parte do hardware, ou seja, adicionar componentes eletrônicos (tais como portas lógicas), modificar ligações entre componentes já pertencentes ao sistema, etc. Suponhamos que temos um sistema com duas entradas e duas saídas, esse simples sistema pode realizar diversas tarefas apenas modificando as instruções que serão carregadas no microprocessador, por exemplo podemos ter o seguinte programa:

- se(entrada 1 ativada)
- {Saída 1 ativada

- Saída 2 desativada}
- se(entrada 1 desativada)
- {Saída 1 desativada
- Saída 2 ativada}

Se o leitor pensar um pouco no processo escrito acima verá que o mesmo pode facilmente ser modificado gerando um total de 14 possíveis processos (estão sendo desconsiderados os processos nos quais todas as saídas estão sempre ativadas e os quais todas as saídas estão sempre desativadas), apenas permutando o estado desativado e ativado para as condições dadas. O mesmo não seria feito tão facilmente no caso de utilizarmos um sistema baseado apenas em Hardware, pois cada simples modificação na lógica modificaria a interligação física dos componentes que formam o sistema de controle.

1.3 O que são CLPs?

CLPs são controladores lógicos programáveis que são utilizados para controlar tarefas que necessitam a performance de PCs industriais mas com uma maior robustez. Estes controladores são multifuncionais, incluindo funções discretas, analógicas e de controle de movimento. Os CLPs possuem uma memória para programas e dados do processo em execução, uma vez que o processo precisa ser modificado, simplesmente pode-se modificar as instruções contidas no programa gravado na memória do CLP.

Para escrever os programas, existem várias linguagens que trabalham diretamente com o CLP, por exemplo, LADDER, e outras linguagens que podem ser utilizadas no desenvolvimento de programas que fazem a Interface Homem Máquina. Por exemplo C++, C#.

Os CLPs tem as seguintes características os colocam entre os melhores controladores para processos industriais:

- Podem trabalhar em ambientes com humidade, altas temperaturas, vibrações e ruídos.
- Possuem interfaces para entradas e saídas já implementadas no controlador
- São facilmente programáveis.

1.4 Componentes do CLP

Um CLP é composto por módulos especializados; Processador, memória, módulos de entrada e saída com interfaces, dispositivos de comunicação, rack, etc. A figura 1.2 mostra um esquema básico de um CLP.

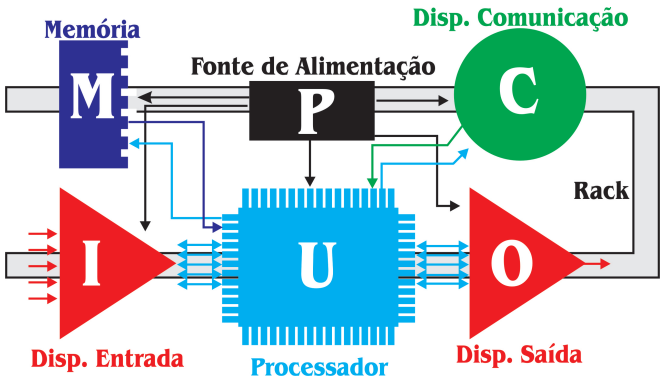

Figura 1.2: CLP - Módulos

Cada módulo mostrado na figura 1.2 é explicado abaixo:

1. Processador: É responsável por ler e interpretar todas as instruções do programa gravado na memória. O processador também o dispositivo que coordena processos de comunicação externa realizadas pelos módulos de comunicação, bem como comunicação entre os módulos presos ao RACK.

2. Fonte de Alimentação: Os módulos do CLP funcionam com uma tensão de 5V, essa tensão não é em geral fornecida por concessionárias de energia elétrica, dessa forma utilizamos esse módulo para converter a tesão fornecida (110 ou 220V) na tensão requisitada pelos módulos.

3. Memória: Local onde são gravados os programas e as variáveis em execução, a memória tem duas áreas reservadas para os estados das entradas e das saídas, chamadas de Matriz de Entradas e Matriz de Saídas, a cada ciclo de execução do programa essas áreas são lidas e se necessário atualizadas.

4. Entradas e Saídas: São as portas por onde o CLP pode se comunicar com o meio exterior, pelas entradas o controlador pode ler os estados do processo que está sendo controlado e pelas saídas pode através de atuadores controlar o processo de acordo com as instruções contidas

no programa em execução. Uma característica bastante interessante dos CLPs é de possuírem as entradas e saídas já acopladas a interfaces isoladoras, de forma que pode-se normalmente ligar dispositivos diretamente as entradas sem a necessidade do desenvolvimento de interfaces, ou seja motores podem ser diretamente ligados as saídas e sensores podem ser diretamente ligados as entradas. As entradas e saídas do CLP podem ser do tipo Analógica ou Digital, e de acordo com o tipo de entrada temos os diferentes tipos de interfaces, são elas EA (entrada analógica), ED (entrada digital), SA (saída analógica), SD (saída digital).

5. Módulos de comunicação: Um CLP pode se comunicar com outros CLPs formando uma rede complexa de controle, essa rede pode conter vários dispositivos não apenas CLPs, esses dispositivos podem se comunicar utilizando diferentes protocolos, para isso são utilizados conversores de protocolos. Alguns protocolos de comunicação dos CLPs e Periféricos são; Profinet, DeviceNet, Ethernet, ASI.

   A figura 1.3 mostra o esquema de uma rede de comunicação.

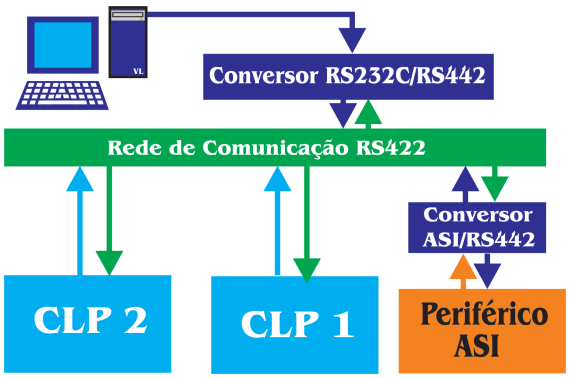

Figura 1.3: Rede de Comunicação

6. Rack: é um apoio onde são encaixados todos os módulos, ao encaixar os módulos os mesmos devem se conectar entre si.

1.5 Arquitetura do CLP

A arquitetura de um CLP é esquematizada na figura 1.4, note que existem 3 tipos de barramentos distintos, um para endereços, um para controle e um outro para dados, a memória também é dividida, em memória de dados e

memória RAM, interessante observar que a memória de programa é alimentada por uma bateria, evitando dessa forma que ocorra a perda do programa em eventuais quedas de tensão.

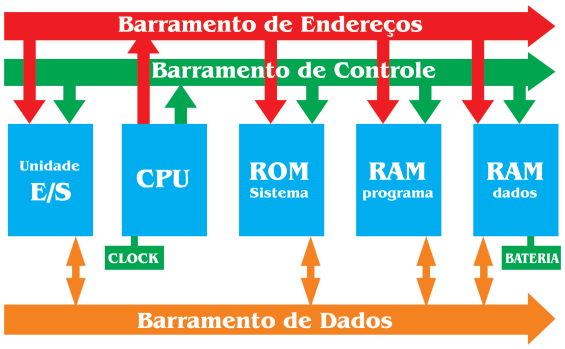

Figura 1.4: Arquitetura do CLP

É explicado a seguir a arquitetura de cada módulo do CLP.

1. CPU: A estrutura interna da CPU depende do microprocessador que a mesma possui, em geral ela é constituída de:

   - Uma unidade lógica aritmética ULA, responsável por operações aritméticas (Soma, Divisão, Subtração e Multiplicação) e operações Lógicas (AND, OR, XOR, NOT).
   - Registradores próximos a ULA para guardar resultados parciais, aumentando a velocidade das operações.
   - Unidade de controle, a qual é responsável por controlar o tempo das operações.

2. Barramento de dados: esse barramento é responsável por trazer dados da memória para serem processados na CPU e de levar resultados dessas operações de volta para a memória, o barramento de dados funciona de forma paralela, ou seja, para trabalhar com dados de 8 bits é necessário um barramento que possua 8 vias e assim por diante, um barramento de dados pode ser serial, no entanto ele apresenta algumas desvantagens em relação ao paralelo; perde-se velocidade e aumenta-se a probabilidade de acontecer erros. E as seguintes vantagens; a fácil implementação e menor custo.

3. Barramento de Endereços: barramento responsável por levar a CPU o endereço do dado que será acessado ou gravado na memória, esse

barramento deve ter o numero de vias de acordo com o tamanho da memória, por exemplo, para uma memória de 256 locações, devemos ter um barramento de 8 vias.

4. Barramento de Controle: responsável por levar sinais da CPU para sincronizar as operações dos demais módulos, por exemplo, se for necessário mostrar os dados contidos em um módulo de memória no barramento, deve ser enviada uma palavra de controle que determine que apenas esse módulo de memória esteja usando o barramento. Os sinais de controle também determinam se os dispositivos de memória vão receber dados das entradas, ou das saídas

5. Memória: Os vários tipos de memória existentes no CLP são listados e explicados abaixo;

   - ROM (memória somente de leitura) Essa memória é não volátil e deve ser programada antes do CLP entrar em execução.
   - RAM (memória de acesso randômico) O CLP possuí dois blocos dessa memória um para dados e outro para programas, essa memória deve ser alimentada por uma bateria visto que a mesma é volátil tendo e tem seus dados perdidos em quedas de tensão. A memória RAM de dados é dividida em blocos, um destinado a guardar uma imagem do estado das entradas do CLP, uma outra guarda o estado das saídas do CLP, outra guarda contadores, temporizadores etc.
   - EPROM (opcional) Essa memória é uma memória apenas de leitura mas que pode ser apagada eletronicamente, mas ainda assim sua regravação não é feita enquanto a mesma é utilizada pelo CLP.

   A capacidade de uma memória é informada da seguinte forma, para uma memória de 256 alocações de 8 bits, temos uma memória 256x8 ou ainda podem ser, especificadas assim, uma memória de 1024 alocações de 8 bits pode ser especificada como Memória de 1Kx8. Observe que para endereça uma memória de 1Kx8 seria necessário um barramento de endereço de 10 vias e para acessar seus dados em paralelo seria necessário um barramento de 8 vias.

6. Unidades de entrada e saída constituem o elo de comunicação entre o CLP e o ambiente ao seu redor, é por meio das unidades de entrada que o CLP pode ler grandezas físicas, tais como temperatura, pressão e umidade, através de sensores. E é por meio das unidades de saída que

pode responder as condições lidas, através de atuadores, como Motores, Válvulas, Luzes etc.

As unidades de entrada e saída possuem varias portas e cada porta possui um endereço único, esse endereço é utilizado quando se está programando o CLP e se deseja acessar uma determinada porta.

As portas de entrada e saída possuem interfaces dotadas de sistemas isoladores, tais como relés e Óptico-isoladores, desse forma torna-se desnecessário a utilização de outros circuitos ao conectar dispositivos a essas portas.

As unidades de entrada podem receber sinais digitais compatíveis (5V) de outros dispositivos, mas podem receber outros sinais, tais como; 24V, 110V e 220V isso devido a interface protetora que o mesmo possui. O sinal de saída de uma unidade de entrada é 5V quando em sua entrada há uma das tensões citadas acima. É mostrado o esquema de uma unidade de entrada na figura 1.5.a

As unidade de saída, recebem uma entrada digital de 5V, e a saída pode ser Digital ou Analógica, como mostra a figura 1.5.b. As unidade de saída são especificadas como sendo, do tipo Relé, Transistore ou Triac, o tipo de unidade a ser usada depende de que atuador será controlado

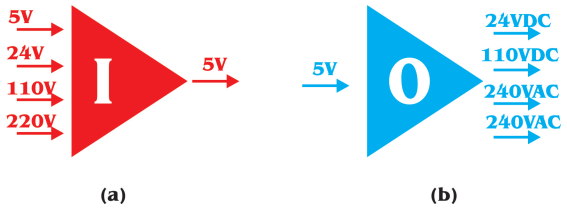

Figura 1.5: Unidades de Entrada(a) e Saída(b)

- RELÉS: São os mais simples, no entanto são relativamente lentos e não devem ser usados para controlar processos que necessitam de uma resposta muito rápida.
- TRANSISTORES: São mais rápidos que os relés mas enquanto podemos usar relés para controlar sistemas AC, e DC, os transistores dever ser usados apenas para controlar cargas DC, e são muito sensíveis a corrente, queimando facilmente com um pico de corrente. Existem os fototransistores que são utilizados em Óptico-Isoladores, nesse caso um LED quando percorrido por corrente libera radiação que ativa o gate do fototransistor e o mesmo passa a conduzir.

- TRIACs: São tão rápidos quanto os Transistores, mas devem ser utilizados apenas para processos AC. São construídos com SCRs a diferença entre eles é que TRIACs trabalham conduzindo em ambas as direções enquanto SCRs só conduzem em uma direção.

1.6 Tipos de CLP

Exitem dois tipos de CLP no mercado, o Single Box e o Modular, o single box, possui todos os módulso em uma única caixa, enquanto o modular possui módulos especializados.

1. Single box: Esse tipo de CLP possui em uma única caixa, Fonte de Alimentação, CPU, Unidades de entrada e saída etc, geralmente possuem 4, 8, 12 ou 24 entradas e 4, 8 ou 16 saídas, possuem uma memória com capacidade entre 300 e 1000 instruções, por serem limitados, esses CLPs são utilizados em pequenas automações, alguns modelos de Single Box, tem a capacidade de aumentar o número de portas de entrada e saída.

2. Modular: Esses CLPs são mais flexíveis e podem ser configurados de acordo com as necessidades do usuário, podem portanto possuir um número muito maior de entradas e saídas e memória quase que ilimitada, os módulos são presos a um Rack e conectados entre si, a ordem na qual os módulos são colocados no Rack não interessa desde que na primeira posição esteja a fonte de alimentação e na segunda esteja a CPU. O esquema de um CLP modular é mostrado na figura 1.6.

Figura 1.6: CLP - Modular

Capítulo 2

Dispositivos de Entradas e Saídas

2.1 Dispositivos de Entrada

2.1.1 Introdução

A primeira parte deste capítulo trata de sensores e transdutores, sensores são dispositivos utilizados para medição de grandezas físicas e transdutores são dispositivos utilizados para a conversão de uma grandeza física em outra, geralmente temos em um mesmo dispositivo o sensor e o transdutor, por exemplo, um dispositivo para medir temperatura possuí um sensor para medir a temperatura e um transdutor que transforma em tensão a temperatura medida. Tratamos também nesse capítulo de dispositivos de saída, que são em sua maior parte, Motores, Relés, Válvulas Solenóides etc.

2.1.2 Sensores

Um sensor é um dispositivo que fornece uma saída em resposta a uma determinada entrada física. Temos por exemplo um termosensor que converte energia térmica em energia potencial. Dispositivos que possuem a capacidade de converter um sinal de uma forma pra outra forma física diferente, são conhecidos como Transdutores. Portanto os sensores geralmente são transdutores, mas existem outros tipos de dispositivos que também são sensores, tais como Motores que convertem energia potencial em energia cinética.

Um sensor que fornece uma saída digital pode ser facilmente ligado às portas de entrada de um CLP , entretanto quando o sensor fornece uma saída analógica devemos antes de conectá-lo ao CLP, converter esse sinal para digital.

Termos que classificam um sensor

- Precisão: Indica até que ponto uma medição pode está errada, por exemplo, se um sensor de temperatura possui uma precisão de +/- 0,1°C e o resultado de uma medida forneceu 20°C o valor real pode ser 19,9°C ou 20,1°C.

- Histerese: Sensores não apresentam uma curva linear da Saída x Entrada, no entanto geralmente para simplificar os projetos é comum supor que essa curva seja linear, essa suposição é uma das fontes de erro entre o valor medido e o valor real, a figura 2.1 mostra a diferença entre valor real e valor medido.

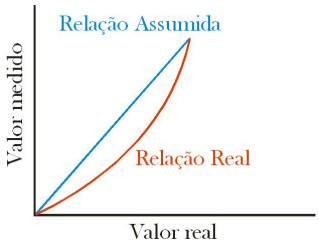

Figura 2.1: Comparação entre Relação Assumida e Real

Uma medição de uma variável física que está crescendo em módulo de um valor X até um valor Y, nos dar uma curva diferente da medição dessa mesma variável se esta tem seu módulo decrescendo de um valor Y até um valor X, essas duas curvas quando combinadas constituem a Histerese do Sensor que está mostrada na figura 2.2.

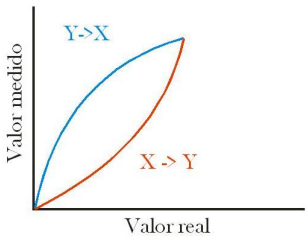

Figura 2.2: Histerese do Sensor

- Faixa de Medição: Define a faixa de valores que o sensor pode medir corretamente e sem danificar-se. Por exemplo, tomarmos um Sensor de

Temperatura que tem uma faixa de -200°C até 800°C, e resolvermos medir uma temperatura de -230°C ele fornecerá um resultado errado e provavelmente próximo de -200°C, caso resolvamos medir uma temperatura de 1000°C certamente vamos queimar o sensor, ou no mínimo queimar algum sistema de segurança que o mesmo possua.

- Tempo de Resposta: Quando a entrada de um sensor muda, ele leva um certo tempo para alcançar um valor fixo e estabilizar, sendo que ele oscila até que alcance esse valor, o tempo de resposta é o tempo decorrido entre o instante que é aplicado o degrau e o instante que a resposta alcança 95% do valor final. O gráfico da resposta a um degrau é mostrado na figura 2.3.

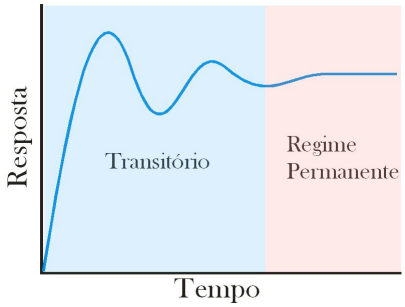

Figura 2.3: Resposta a um degrau

- Tempo de Subida: se refere ao tempo que leva para a saída subir de 10/100 do valor de regime permanente para 90/100 desse valor.

- Sensibilidade: indica quanto a saída mede para cada unidade da entrada, a fração saída/entrada igual a 20uV/°C nos diz que a cada aumento de 1° da entrada a saída aumenta de 20uV (estamos novamente supondo a linearidade).

- Estabilidade: é a habilidade de um sensor dar a mesma saída quando uma mesma entrada é aplicada durante um certo tempo. O termo Drift é frequentemente usado para descrever a mudança que ocorre na saída após um certo tempo que temos uma entrada constante. O termo Zero Drift é usado para as mudanças que ocorrem na saída quando na entrada temos um zero.

- Repeatibilidade: É definido como sendo a habilidade de um sensor dar o mesmo resultado para repetidas medições de um mesmo valor.

- Confiabilidade: É definido como a probabilidade de um sistema de medição (sensor) trabalhar em um nível aceitável de performance, por um determinado período, sujeito a determinadas condições ambientais.

2.1.3 Interruptores

São sensores usados para gerar sinais de (ON/OFF).

Interruptores Mecânicos

Os interruptores mecânicos podem funcionar de duas formas:

- Modo (A) (lógica negativa)
  1. Interruptor ON saída em nível baixo
  2. Interruptor OFF saída em nível alto
- Modo (B) (lógica Positiva)
  1. Interruptor ON saída em nível alto
  2. Interruptor OFF saída em nível baixo

Os Interruptores estão disponíveis como:

- Normalmente Abertos (NO)
- Normalmente Fechados (NC)

Alguns exemplos de interruptores são mostrados na figura 2.4.

Figura 2.4: Exemplos de Interruptores

Interruptores de Proximidade

Esses sensores são usados para detectar a presença de algum objeto ou alguém sem a necessidade de contato físico, existem muitas formas de projetar tais interruptores, porém algumas formas só funcionam para objetos metálicos. Abaixo é explicado o funcionamento de alguns desses interruptores:

- Eddy Current Switch: Esse tipo de sensor possui uma bobina energizada por uma corrente alternada, essa corrente cria um campo magnético alternado, quando um objeto metálico se aproxima da bobina, correntes são induzidas no objeto criando outro campo magnético, esse segundo campo cria uma f.e.m. na bobina, como resultado a voltagem necessária para manter a corrente constante na bobina muda, essa voltagem então é medida de acordo com a proximidade dos objetos, podemos então usar essa voltagem para ativar uma chave eletrônica. Esse tipo de sensor detecta objetos a uma distância de 0.5 a 20mm. A figura 2.5 mostra o esquema de um Eddy Current Switch.

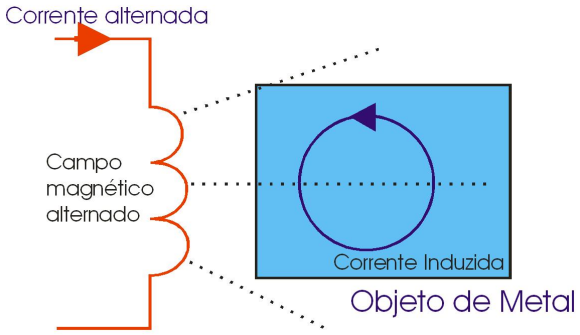

Figura 2.5: Eddy Current Sensor

- Capacitive Proximity Switch: esses interruptores podem ser usados para detectar Objetos metálicos e não metálicos. A capacitância de um par de placas separadas por uma distância, depende da distância que estão separadas, sendo que quanto menor a distância, maior a capacitância. Quando esse tipo de sensor é usado para detectar objetos não metálicos a capacitância do capacitor vai depender diretamente do dielétrico entre suas placas, e nesse caso as placas são, o sensor e a terra, sendo que o objeto não metálico é o dielétrico.

- Reed Switch: Nesse caso temos um invólucro de vidro contendo duas tiras de um material ferromagnético muito próximas mas sem se tocar, quando um material magnético se aproxima as tiras se magnetizam e atraem uma a outra fechando portanto o contato. Esse tipo de chave só fecha quando o objeto magnético está muito próximo, cerca de 1mm. O esquema do Reed Switch é mostrado na figura 2.6.

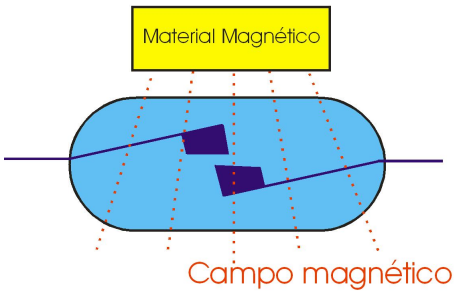

Figura 2.6: Reed Switch

Interruptores Fotoelétricos

Esses dispositivos podem operar de duas formas, como "transmissive types"onde o objeto que está sendo identificado bloqueia o feixe de radiação, e como "reflexive types"quando o objeto que está sendo identificado reflete o feixe de radiação, os esquemas desses interruptores são mostrados na figura 2.7.

- No primeiro tipo (figura 7.A) temos um emissor, geralmente de radiação infravermelha e em frente ao emissor temos o receptor, quando um objeto passar por entre esses dispositivos ele bloqueia o feixe de radiação do emissor e então o sensor acusa a presença de um objeto.

- O segundo tipo (figura 7.B) temos emissor e receptor orientados para o mesmo lugar, quando um objeto posiciona-se a frente do emissor a radiação é refletida voltando então ao receptor daí o sensor acusa a presença de um objeto.

Encoders

O termo encoders é usado para dispositivos que fornecem uma saída digital para um deslocamento angular ou linear. Um dispositivo muito conhecido

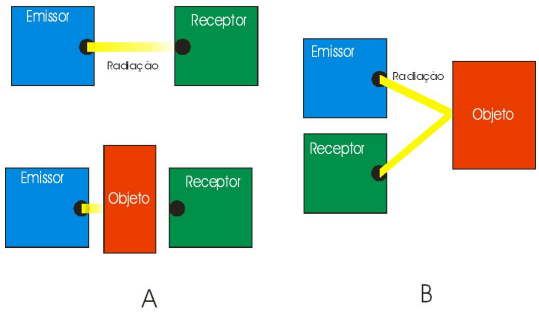

Figura 2.7: Interruptores Fotoelétricos (A) Transmissive, (B) Reflexive

que usa dois enconders é o mouse tradicional o qual possui uma "bolinha"em sua base inferior.

- O encoder de deslocamento linear, como os usados por mouses tradicionais, funciona da seguinte forma: Um LED emite um feixe de luz por entre aberturas em um disco, ao atravessar o disco o feixe atinge um foto-transistor ou um foto-diodo. Ao girar o disco o feixe emitido pelo LED por vezes é bloqueado e por vezes é deixado passar por entre as aberturas do disco, como resultado temos uma seqüência de pulsos na entrada do foto-diodo ou do foto-transistor, o mesmo fornece como saída uma seqüência de ON/OFF. Um hardware/software externo deve então entender o que quer dizer o chaveamento e tomar a decisão sobre que fazer, a figura 2.8 mostra o esquema de um encoder usado em mouse.

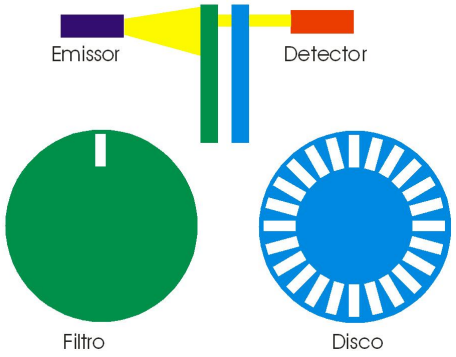

Figura 2.8: Forma básica de um Encoder Incremental

Podemos ainda usar essa dispositivo para fazer um encoder de deslo-

camento angular relativo, o qual a cada bloqueio/passagem do feixe gera um incremento ou decremento do ângulo atual, dessa forma se tivermos um disco com 60 aberturas podemos ter um encoder de deslocamento angular com uma resolução de 360/60 ou seja de 6º, em outras palavras, a cada bloqueio/passagem do feixe temos um incremento ou decremento de 6º.

- O encoder de deslocamento angular absoluto, esse encoder se diferencia do anterior por ter em seu disco diferentes regiões que definem uma posição única, imagine em vez de um simples disco, temos quatro discos concêntricos gerando um disco mais complexo, e temos 4 sensores de forma que podemos definir 16 possíveis regiões desse conjunto de discos da seguinte forma: De acordo com qual conjunto de sensores está recebendo o feixe do LED, considere que quando o feixe atinge o sensor temos "1"e quando o feixe é bloqueado temos "0"podemos então configurar os discos de forma que partamos de 0000 até 1111, ou seja de 0 a 15 totalizando as 16 posições, caso nosso sistema possua 5 discos e 5 sensores podemos ter usando a mesma lógica 0 a 31 ou seja 32 posições. Um esquema de um encoder absoluto com 8 posições é mostrado na figura 2.9.

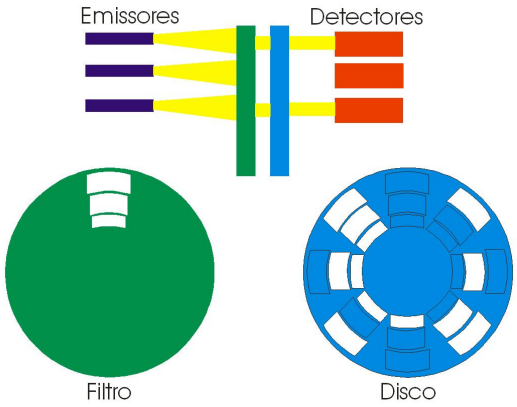

Figura 2.9: Encoder Absoluto com 8 Posições

2.1.4 Sensores de Temperatura

Um simples sensor de temperatura que gera um sinal de ON/OFF quando uma dada temperatura é alcançada, pode ser feito com duas finas tiras de

metais que possuem coeficiente de expansão térmica diferentes coladas uma na outra, de modo que quando a temperatura aumenta as mesmas se expandem de forma desigual gerando portanto uma curvatura na sua forma, essa curvatura pode ser usada pra fechar ou abrir uma chave, na figura 2.10 temos o esquema de um sensor desse, ligado a uma chave, quando está aberto (figura 2.10.a) e fechado (figura 2.10.b).

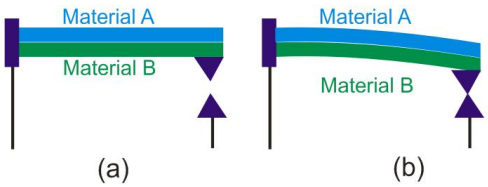

Figura 2.10: Chave aberta (a) Chave fechada (b)

Outros tipos de sensores de temperatura bastate utilizados são mostrados a seguir:

- Detector Resistivo de Temperatura (DRT) Esse tipo de dispositivo tem como seu principio de funcionamento a propriedade que os metais tem de mudar sua resistência em função da temperatura, dentre os metais alguns elementos se destacam nessa propriedade, são eles: Platina e o Níquel, os quais variam sua resistência de maneira linear com a temperatura através de uma ampla faixa de temperatura. O DRT pode ser utilizando em uma configuração de Ponte de Wheatstone (figura 2.11.a) ou um simples divisor de tensão (figura 2.11.b).

- Termodiodos e Termotransistors, uma característica desses dispositivos é que a taxa com a qual os elétrons e lacunas se difundem através das junções é afetada pela temperatura. Circuitos integrados estão disponíveis no mercado, os quais combinam um termodiodo ou termotransistor com um circuito que dá uma saída em voltagem. Um CI largamente usado é o LM35 (figura 2.12) o qual da uma saída de 10mV/^{o}C, O LM35 que tem uma saída analógica, pode ainda ser combinado com um comparador produzindo uma saída lógica para dada temperatura de referência, o comparador utilizado pode ser o LM3911N o qual vai controlar a função ON/OFF do sensor.

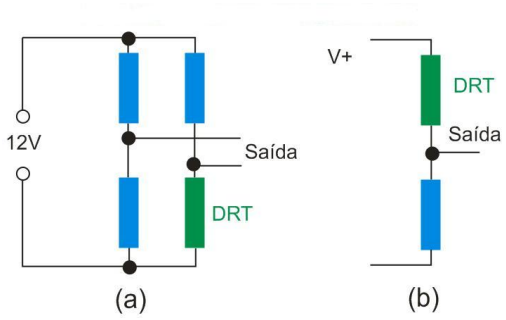

Figura 2.11: Ponte de Wheatstone (a) Divisor de Tensão (b)

Figura 2.12: LM35

2.1.5 Sensores de Posição e Deslocamento

O termo sensor de posição é usado para um dispositivo que da a medida da distância entre um ponto de referência e o ponto atual, já o termo sensor de deslocamento é usado para um dispositivo que da a medida da distância entre o ponto atual e um ponto previamente guardado. Sensores resistivos de posição linear e angular são amplamente usados e relativamente baratos. Esses sensores são também chamados de Potenciômetros Lineares e Rotativos. Um exemplo de Sensor de posição resistivo é o potenciômetro linear o qual pode ter em uma das suas entradas uma tensão "V"na outra entrada o terra, ao mover o potenciômetro seja girando ou deslocando temos na saída uma tensão "Vo"analógica de módulo compreendido no intervalo "0"a "V". Outros tipos desses sensores são explicados abaixo:

- "Linear Variable Differential Transformer"(LVDT) esse sensor fornece uma voltagem na saída relativa a posição de uma haste de ferro. O LVDT é constituído por 3 bobinas simetricamente colocadas através da haste de ferro móvel. Uma das bobinas ocupa toda a extensão da haste e caracteriza o terminal de entrada do dispositivo, que recebe uma corrente alternada, na saída temos as outras duas bobinas cada

uma ocupando aproximadamente metade da extensão da haste de ferro. Quando a haste de ferro não está deslocada temos que a tensão V1 = V2, quando a haste é delocada é então criada uma diferença de potencial V1-V2=V3 a saída V3 que é AC é então convertida para DC amplificada daí pode ser mandada ao CLP, o esquema desse dispositivo é mostrado na figura 2.13.

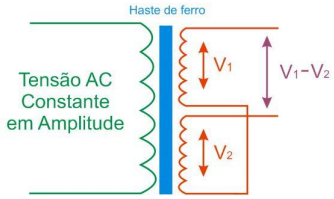

Figura 2.13: LVDT

- Um outro dispositivo que pode ser usado como sensor de deslocamento é um capacitor de placas planas e paralelas móveis, sendo que a capacitância ira mudar quando as placas tiverem a distancia entre si alterada ou se moverem paralelamente uma a outra.

2.1.6 Sensores de Deformação (Strain Gauges)

Quando um material é deformado sua resistência é alterada, a fração de mudança na resistência é proporcional a fração de mudança no comprimento do material. Por exemplo, se temos um fio metálico de resistência R e o mesmo sofre uma deformação Δ R então temos a seguinte expressão:

$$\frac{\Delta R}{R} = GD$$

, onde "G"é chamado de Gauge Factor e "D"é a deformação. Para metais o gauge factor é de aproximadamente 2, enquanto para semicondutores chega a 100. Os Sensores de Deformação são geralmente acoplados a uma fina camada de plástico ou filme. Para medir a variação de resistência de um sensor de deformação podemos usar uma ponte de Wheatstone, na qual um dos resistores é substituído pelo Sensor de Deformação, encontramos entretanto o problema de que, sensores de deformação são sensíveis também a temperatura, logo devemos colocar do lado oposto da ponte de Wheatstone uma resistência com sensibilidade semelhante a temperatura, para assim compensar a variação da resistência devido a temperatura.

2.1.7 Sensores de Pressão

Esses sensores geralmente são produzidos utilizando um diafragma, esse diafragma consiste de um fino disco de plástico ou metal, preso pelas extremidades e funciona da seguinte forma, quando existe uma diferença de pressão entre os dois lados do diafragma o mesmo se deforma, de forma que a deformação é proporcional a diferença de pressão, podemos usar um Sensor de Deformação para medir essa deformação, ou podemos usar um capacitor de placas planas paralelas móveis, o qual teria sua capacitância mudada em função da deformação do diafragma, ou ainda usando a deformação para comprimir um cristal piezoelétrico o qual quando submetido a uma pressão, gera um campo elétrico (em um eixo transversal àquele onde foi aplicado a pressão) que pode ser coletado como tensão. Esse cristal é bastante utilizado em circuitos eletrônicos para se gerar o clock de Trigger em certos componentes síncronos do circuito, como contadores, registradores etc. Ele é utilizado por exemplo, para fazer os relógios de pulso, em que é necessário uma precisão muito boa para se mostrar as horas, minutos e segundos. Também conhecido por estudantes de Engenharia Eletrônica pelo termo técnico XTAL. Um exemplo desse tipo de sensor é o MPX100AP da Motorola o qual foi construído em vácuo em um lado do diafragma então a deformação do diafragma da a medida da pressão absoluta aplicada ao outro lado do diafragma, esse sensor fornece uma saída em voltagem proporcional a pressão aplicada, com uma sensibilidade de 0.6mV/kPa, Um esquema simples de um sensor de pressão é mostrado na figura 2.14.

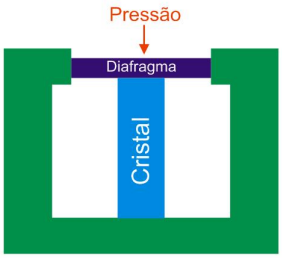

Figura 2.14: Sensor de Pressão

2.1.8 Detectores de Nível de Água

Sensores de pressão podem ser usados para monitorar a profundidade de um líquido em um tanque. Uma vez que a pressão de uma coluna de líquido

pode ser calculada por:

$$\Delta P = hg\rho$$

Onde "h"é a nível da água, "g"é a gravidade e "ρ"é a densidade do líquido, o esquema desses sensores de nível de água é mostrado nas figura 2.15.

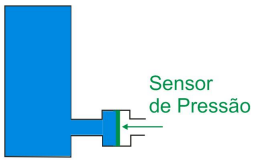

Figura 2.15: Sensor de nível de água

Ás vezes é desejado apenas detectar que o nível de um líquido chegou a uma altura X, dessa forma podemos usar uma chave flutuando na água, a qual acompanha o nível da mesma, e outra chave fixa, o nível da água sobe fazendo com que as placas se toquem, daí um sinal de curto pode ser enviado pra um circuito externo e este envia um sinal de ON/OFF.

2.1.9 Medida de escoamento de fluídos.

Uma forma comum de fazer essa medição é medir a diferença de pressão entre dois pontos do trajeto do fluído em um tubo. Devido as perdas de carga um fluído diminui a pressão que exerce sobre as paredes do tubo ao longo de sua trajetória, essa diferença de pressão pode então ser monitorada por um sistema de Diafragmas conectados a Sensores de Deformação, o esquema básico desse sistema é mostrado na figura 2.16.

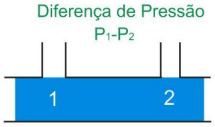

Figura 2.16: Sensor de escoamento

2.1.10 Sensores Inteligentes

O termo Smart Sensor é usado para um sensor que possui em um só chip uma unidade de processamento, conversores de dados, e uma memória não

volátil, tal como uma EEPROM, a memória não volátil evita que alguns dados importantes sejam perdidos quando ocorre um corte no fornecimento de tensão. O Padrão IEEE 1451.4 dita as regras para fazer a interface entre sensores inteligentes e dispositivos de saída, esse padrão visa permitir a fácil instalação de transdutores analógicos a sistemas de medição digital. Esse padrão requer ainda uma memória não volátil a qual guardará dados de comunicação, permitindo a instalação de dispositivos plug-and-play.

2.2 Dispositivos de Saída (Atuadores)

2.2.1 Introdução

As saídas do CLP são geralmente protegidas por Relés, SCRs, Transistores ou Triacs, o tipo de isolamento vai depender de qual dispositivo se deseja conectar a saída, por exemplo, no caso de um dispositivo de saída que será usado em um processo que requer uma reposta rápida a uma entrada, não se deve utilizar um relé visto que o tempo de resposta desse isolador é relativamente grande quando comparado a transistores, SCRs ou Triacs. O relé que é o mais simples dentre esses componentes isoladores, é composto por uma bobina que quando percorrida por uma corrente na entrada fecha uma chave na saída, seu esquema é mostrado na figura 2.17. Nos próximos tópicos serão apresentados alguns dos dispositivos de saída.

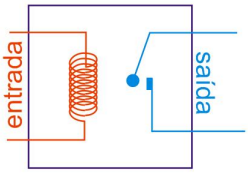

Figura 2.17: Esquema de um Relé

2.2.2 Válvulas de Controle de Direção (VCD)

Podemos utilizar um solenóide para agir como um atuador, do tipo Válvula Solenóide, esse tipo de válvula geralmente possuí duas posições, uma posição é conseguida quando não há corrente atravessando o solenóide, posição (B), a outra posição quando há corrente atravessando o solenóide, posição (A). para passar da posição (B) para posição (A) fazemos passar uma corrente no solenóide, o processo de volta, da posição (A) para posição (B) é conseguido

mediante a força de uma mola, que apesar de ser de ter menor módulo que a força magnética do solenóide, é capaz de trazer de volta o pistão para posição (B) quando é cessada a corrente no solenóide. O esquema desse tipo de válvula é mostrado na figura 2.18.

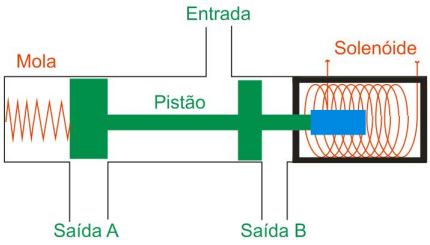

Figura 2.18: VCD

Os símbolos para representar esse tipo de válvulas são mostrados na figura 2.19, é mostrado o simbolo de uma válvula genérica de duas posições na figura 2.19.a, na figura 2.19.b é mostrado o símbolo de uma válvula de duas posições, na posição (1) o fluído segue de (A) para (T) e de (P) para (B), enquanto que na posição (2) o fluído segue de (P) para (A) e de (B) para (T).

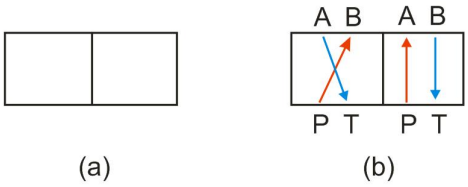

Figura 2.19: Simbolos

Nos diagramas de um projeto que utiliza válvulas de direção, o método usado para mudar a direção de escoamento deve ser mostrado no símbolo da válvula, alguns desses métodos são mostrados na figura 2.20, em (a) Solenóide, (b) Botão, (c) Mola

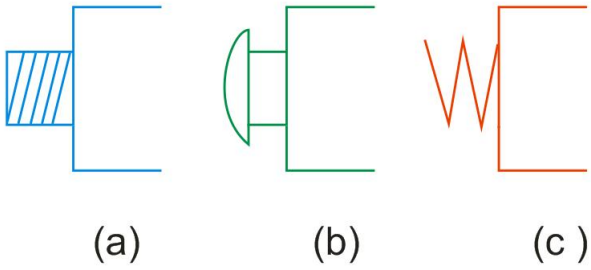

Figura 2.20: Simbolos modo de atuação

2.3 Motores

Motores DC

Motores DC são constituídos bobinas enroladas em um material ferromagnético, esse conjunto é chamado de armadura, a armadura é montada de forma que seja possível a mesma rotacionar, por isso é chamada também de rotor. Para a armadura desenvolver uma rotação uma corrente deve passar pela a bobina da armadura e essa bobina deve estar imersa em um campo magnético, esse campo pode ser criado por um circuito chamado estator, ou por um imã permanente. Para que o movimento de rotação do motor não cesse, é utilizado um conjunto de escovas que fazem a comutação das bobinas de forma a evitar que a bobina se alinhe ao campo magnético, e preservando então a rotação. Um esquema simples de um motor apenas com uma bobina é mostrado na figura 2.21, observe que a velocidade de rotação deve ser grande o bastante para evitar que o motor pare no momento de comutação dos terminais da bobina.

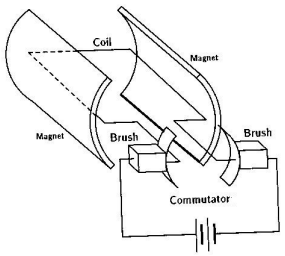

Figura 2.21: Simples Motor DC

Controle de Velocidade do Motor DC

O controle da velocidade de um motor DC pode ser realizado aplicando um sinal PWM a sua entrada, e variando a largura do pulso, o CLP pode ser usado para controlar a largura do pulso e assim controlar a velocidade de rotação de um motor DC, na figura 2.22 é mostrado o esquema de um PWM, observe que dependendo da largura do pulso o motor enxerga diferentes valores de tensão em sua entrada.

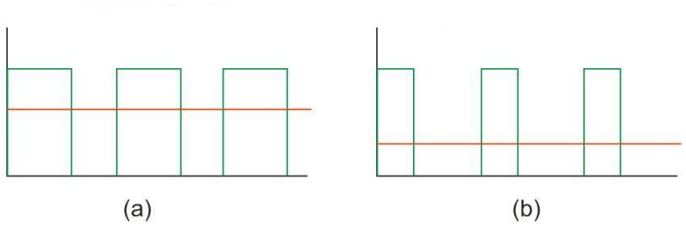

Figura 2.22: PWM, linha vermelha informa tensão média

2.3.1 Motores de Passo

Esses motores produzem um movimento de rotação de grande precisão através de um incremento na sua posição atual de um ângulo X, quanto menor o valor desse ângulo maior a precisão do motor, existem três tipos de motores de passo.

Os motores de passo são classificados ainda como Unipolar e Bipolar, nos unipolares temos que a corrente em dada bobina só flui em um sentido, enquanto no bipolar pode fluir em ambos sentidos.

Relutância Variável

Tem um rotor com várias polaridades feito com ferro doce e um estator laminado. Eles geralmente operam com ângulos de passo de 5 a 15 graus. Na Figura 2.23, quando fase A é energizada, quatro dentes de rotor se alinham com os quatro dentes do estator da fase A através de atração magnética. O próximo passo é dado quando a fase A é desligada e fase B é energizada fazendo o rotor girar 15 graus à direita. Continuando a seqüência, a fase C é energizada e depois a fase A novamente.

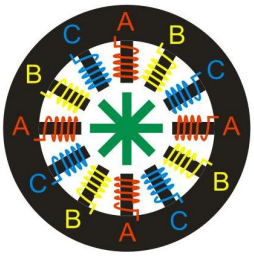

Figura 2.23: Motor de passo tipo: Relutância Variável

Magnético Permanente

Motores de magnético permanente diferem dos de relutância variável pois têm rotores de material alnico ou ferrite sem dentes. Energizando as quatro fases em seqüência, o rotor gira, pois é atraído aos pólos magnéticos. O motor mostrado na Figura 2.24 dará um passo de 90 graus quando os enrolamentos ABCD forem energizados em seqüência.

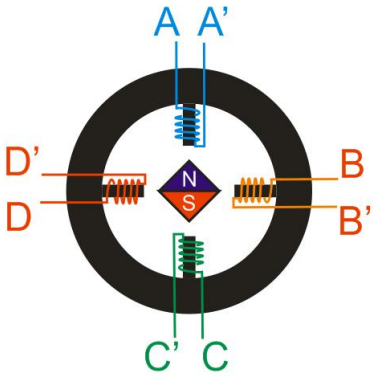

Figura 2.24: Motor de passo tipo: Magnético Permanente

Driver para Motor de Passo

Para simplificar o controle de motores de passo, existem no mercado CIs que funcionam recebendo na sua entrada uma seqüência de pulsos, essa seqüência pode facilmente ser gerada por um CLP, e retornam como saída a seqüência correta para a rotação de um motor de passo, a figura 2.25 mostra como seria a entrada e a saída de um driver desses.

Um CI bastante utilizado é o SAA1027, que disponibiliza um driver para o controle de motores de passo unipolar de 4 fases. Outros CIs como o

ULN2003 são largamente usados no controle desses motores.

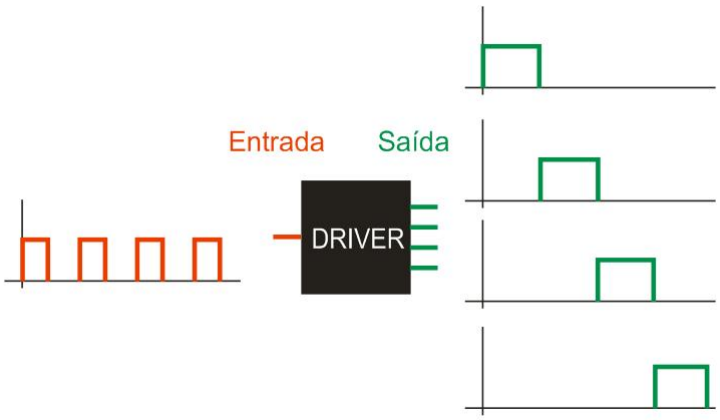

Figura 2.25: Driver

Capítulo 3

Processamento de Entradas e Saídas

3.1 Introdução

As unidades de entrada e saída são os meios de comunicação do CLP com o ambiente a sua volta, essas unidades possuem interfaces que as protegem isolando-as fisicamente dos dispositivos de entrada e saída. Será explicado nesse capitulo a forma como devem ser conectados os dispositivos de entrada e saída ao CLP.

3.2 Dispositivos de Entrada e Saída

Os dispositivos de entrada fornecem sinais de vários tipos, assim como os dispositivos de saída ou atuadores necessitam dos mesmos sinais para seu funcionamento. Esses sinais são classificados da seguinte forma:

- Analógicos: Um sinal que fornece todos os valores entre seu máximo e mínimo, assim sendo, se tivermos um sinal analógico que varia entre 1V e 2V, esse sinal para ir de um extremo ao outro deve passar por todos os inúmeros valores ou seja, 1,0000001, 1,0000002 etc.
- Discretos: Quando o sinal vária apenas entre um estado ON e um estado OFF, ou seja 5V e 0V respectivamente.
- Digitais: um sinal digital pode ser entendido como uma seqüência de pulsos.

Apesar dos dispositivos de entrada trabalharem com valores variados de sinais, a CPU geralmente deve receber apenas um sinal digital, entre 0 e 5V,

assim como a saída da CPU é um sinal digital entre 0 e 5V, as interfaces permitem que entradas maiores sejam transformadas na tensão suportada pela CPU, assim como transforma a saída da CPU para a tensão necessária a atuadores conectados as portas de saída.

3.3 Unidades de Entrada

Existem duas maneiras nas quais um dispositivo de entrada pode ser conectado, chamadas de "sourcing"e "sinking"essa classificação depende de como o dispositivo é conectado a porta, no modo sourcing, (figura 3.1.a) o dispositivo é a unidade de entrada fornece a corrente ao dispositivo, no modo sinking, (figura 3.1.b) o dispositivo de entrada fornece a corrente a unidade de entrada.

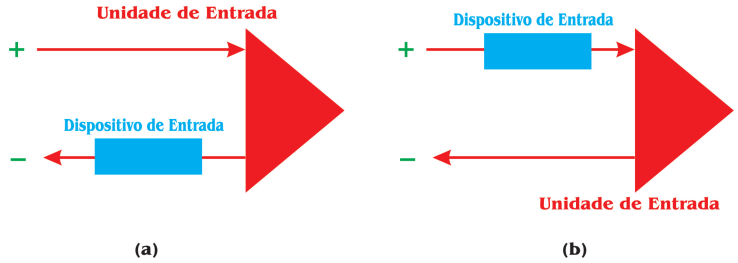

Figura 3.1: (a) Sourcing, (b) Sinking

Os circuitos isoladores são utilizados para proteger a CPU de picos de tensão ou tensões elevadas. Os circuitos básicos de isolamento para tensões de entrada DC e AC são mostrados nas figura 3.2 e figura 3.3.

O circuito isolador de entradas DC possui um LED que informa quando há um sinal de entrada da unidade, e um Isolador Óptico, um resistor é colocado entre a tensão V e a entrada da CPU para evitar correntes altas.

O circuito isolador de entradas AC é também dotado de um sistema que indica que há um sinal na unidade, ele também é constituído por uma ponte a qual é utilizada para retificar o sinal, após retificado o sinal passa por um isolador. Na entrada da CPU é colocado um capacitor esse capacitor após carregado fornece a CPU uma tensão DC.

3.4 Multiplexadores

Vários sinais analógicos podem ser coletados por um único canal de entrada do CLP, isso é conseguido com a utilização de multiplexadores, um

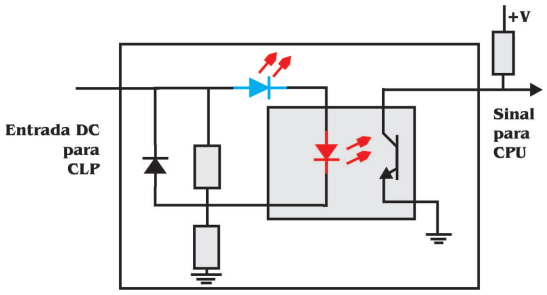

Figura 3.2: Interface DC

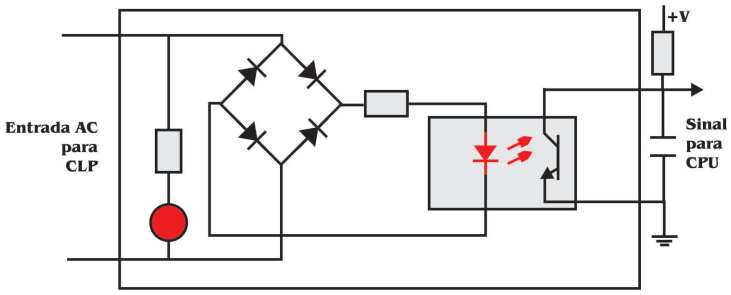

Figura 3.3: Interface AC

multiplexador é um dispositivo que permite que um canal possa ser compartilhado por vários dispositivos, sendo que cada um o utiliza durante um intervalo de tempo.

Por exemplo, um multiplexador 8:1 possui 8 entradas que vão compartilhar uma única saída, para definir qual entrada usará a saída é necessário que existam sinais de controle, para 8 entradas, é necessário uma palavra de controle de 3 bits.

Um esquema de um multiplexador é mostrado na figura 3.4.

Figura 3.4: Multiplexador 8:1

3.5 Conversor A/D

A maioria dos sensores existentes no mercado são analógicos, no entanto as vezes é necessário trabalhar com esses dados de forma digital, daí vem a necessidade de conversores A/D, esses conversores, recebem em sua entrada um sinal analógico e como saída fornecem o valor digital.

Figura 3.5: Conversor A/D de 8bits

A figura 3.6 mostra o esquema de uma conversão A/D utilizando um conversor de 3bits.

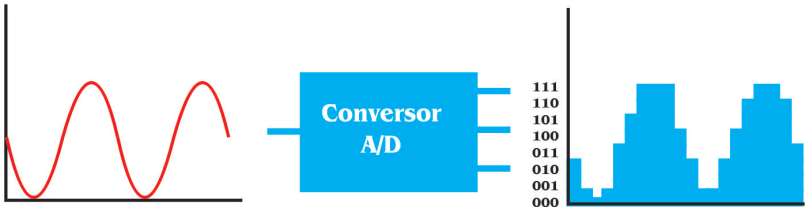

Figura 3.6: Conversão A/D 3bits

Um parâmetro importante dos Conversores A/D é a resolução quanto maior, mais preciso ele se torna o conversor, por exemplo, se um conversor A/D de 8 bits é utilizado para converter um sinal analógico de valores entre 2V e 12V ou seja um intervalo de 10V então ele terá uma resolução de 10/255 = 0,04V, dessa forma a cada variação de 40mV na entrada o sinal na saída é incrementado em 1.

Um esquema de um conversor A/D é mostrado na figura 3.5.

3.6 Conversor D/A

Outro conversor muito utilizado é o conversor Digital-Analógico, esse conversor recebe em sua entrada formada por n bits, um sinal do controlador. O controlador envia um numero binário de 8 bits e o conversor o transforma em

tensão analógica, nesses conversores é necessário pinos para informar quais as tensões de referência, por exemplo caso seja desejado ter valores entre 6V e 8V, coloca-se na entrada REF(-) uma tensão de 6V e na entrada REF(+) uma tensão de 8V, dessa forma quando na entrada do conversor de 8bits tivermos 255, a saída será 8V, se tivermos 0 na entrada a saída será 6V, os outros valores de entrada geram valores entre 6V e 8V na saída.

Um esquema de um conversor D/A é mostrado na figura 3.7

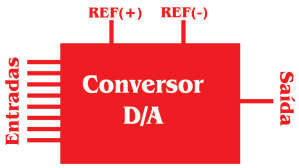

Figura 3.7: Conversor D/A

3.7 Unidades de Saída

Os dispositivos de saída podem ser conectados as unidades de saída de duas formas Sinking e Sourcing similar a forma como foi mostrado para os dispositivos de entrada, as unidades de saída são de 3 tipos, Relés, Transistores e Triacs.

3.7.1 Saídas do tipo Relé

Como já foi explicado esse tipo de saída é utilizado para controlar processos que não necessitem de uma resposta muito rápida, uma vez que o chaveamento de Relés é um processo mecânico. A figura 3.8 mostra o esquema desse tipo de saída.

É importante observar que todas as saídas possuem um fusível para sua proteção contra correntes altas.

3.7.2 Saídas do tipo Transistores

Utilizados em processos que necessitam de um rápido chaveamento, no entanto só trabalham com atuadores DC. É mostrado na figura 3.9 o esquema desse tipo de saída.

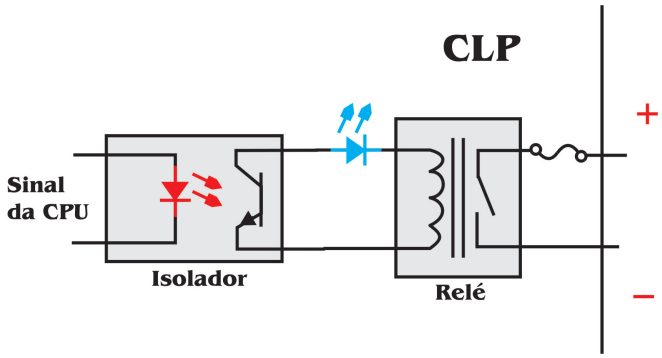

Figura 3.8: Saída tipo Relé

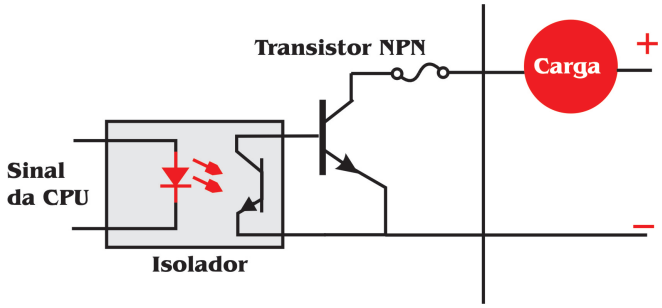

Figura 3.9: Saída tipo Transistor

3.7.3 Saídas do tipo Triac

Similar aos Transistores mas só trabalham com atuadores AC. É mostrado na figura 3.10 o esquema desse tipo de saída.

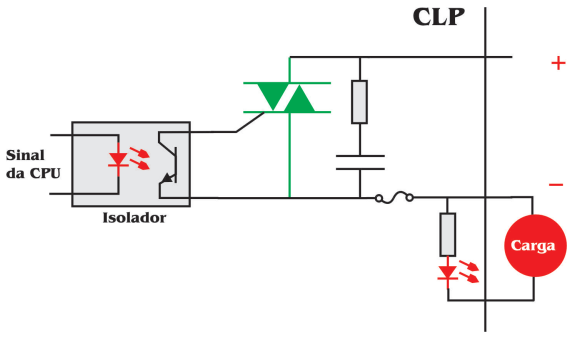

Figura 3.10: Saída tipo Triac

3.8 Tratamento dos Sinais

Sensores conectados ao CLP nem sempre dão como resposta sinais que podem ser lidos pelas entradas do CLP, por essa razão faz-se necessário um tratamento desses sinais. Tomemos como exemplo um sensor que fornece em sua saída valores entre 4mA e 20mA, pode-se utilizar um resistor de 500 Ohms, e fazer passar por ele a corrente de saída do sensor, tendo então um sinal que varia de 2 a 10V. Esse sinal deve então ser lido pelo CLP através de um conversor A/D e daí dependendo do valor lido o usuário define no programa o que será feito.

Outras vezes o sinal de entrada é muito grande, então o tratamento que deve ser feito é passar esse sinal por um simples divisor de tensão, por outro lado quando o sinal é muito pequeno, o que devemos fazer é amplificar esse sinal, utilizando Amplificadores Operacionais, os esquemas de tratamento de sinais utilizando divisor de tensão e Amp-Op são mostrados na figura 3.11.

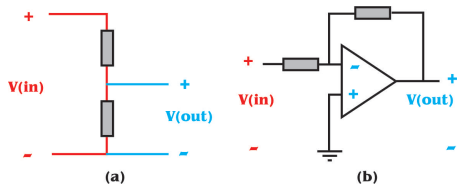

Figura 3.11: (a) Divisor de Tensão, (b) Amplificador

3.9 Conexões Remotas

Quando em um sistema existem muitos dispositivos situados a uma longa distância do CLP, é aconselhável utilizar módulos para uma conexão remota, da seguinte forma, um módulo de entradas/saídas é conectado ao CLP e todos os dispositivos distantes ao CLP são conectados a esse módulo. Um esquema dessa técnica é mostrado na figura 3.12.

Em outras ocasiões é necessário devido ao grande número de dispositivos de entrada e saída, e a esses dispositivos estarem muito espalhados, utilizar uma rede de CLPs, onde um CLP chamado Master coordena as atividades dos demais chamados Escravos.

Os cabos utilizados para comunicação entre CLPs e Módulos Remotos e CLP Máster e CLPs Escravos podem ser Cabos Coaxiais, Fibra Óptica etc. A escolha do cabo para a comunicação será função da quantidade de informação que deverá ser transmitida pelo mesmo.

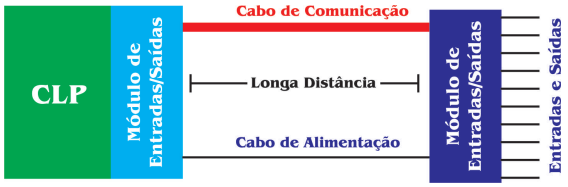

Figura 3.12: Esquema de uma conexão remota

3.10 Protocolos de Comunicação

Os protocolos de comunicação mais conhecidos são, comunicação Serial e comunicação Paralela, na Serial temos que um dado é transmitido bit a bit, um de cada vez, dessa forma se temos um dado de 8 bits, o mesmo será transmitido em 8 partes. A comunicação paralela ocorre de forma diferente, nesse caso todos os bits do dado são transmitidos de uma vez só, isso torna o meio de comunicação mais caro, mas ao mesmo tempo torna a comunicação mais rápida. A comunicação Serial é utilizada para transmitir dados em longas distâncias, pois é mais seguro e econômico construir um longo cabo para comunicação serial do que construir 8, 16 ou até 32 cabos para comunicação paralela. A comunicação paralela é utilizada geralmente em curtas distâncias onde se necessita de grande velocidade, internamente os CLPs trabalham com comunicação paralela, dessa forma é necessário uma conversão dos dados que chegam de forma Serial, essa conversão é conseguida através de uma UART (Universal Asynchronous Receiver-Transmiter) colocada nas unidades de entrada e saída. Uma analogia entre comunicação serial e paralela pode ser feita com o Pistas com várias faixas e com apenas uma faixa, a analogia é esquematizada na figura 3.13.

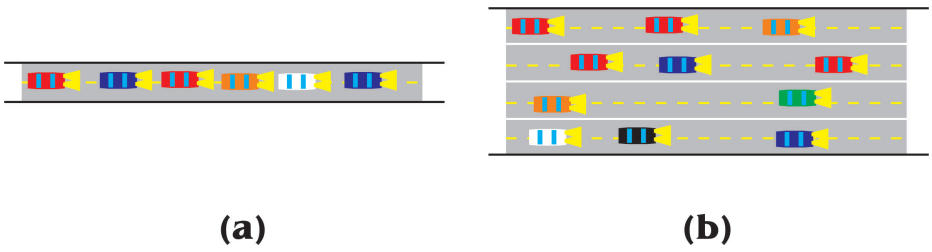

Figura 3.13: Analogia entre comunicação e tráfego

Nota-se que na figura 3.13a que representa a comunicação serial o número de bits(carros) transmitidos em um dado intervalo de tempo, é muito menor do que na comunicação paralela, representada pela figura 3.13b.

3.10.1 Padrões de Comunicação Serial

Para que a comunicação serial ocorra com sucesso, é necessário que algumas especificações sejam feitas, dentre elas temos:

- Qual tensão representará 1 e qual representará 0.
- Ser capaz de determinar onde inicia e termina uma palavra, visto que várias palavras serão enviados pelo mesmo cabo.
- A velocidade da conexão em bits por segundo.
- Sincronizar os clocks a cada fim de transmissão.
- Protocolos de Comunicação, esses protocolos são usados para determinar quando um receptor está pronto para receber uma mensagem e para determinar quando o emissor está pronto para enviar, isso é geralmente conseguido incluindo 2 bits novos a palavra, um para informar que o emissor está pronto e outro para informar que o receptor está pronto.
- Error-checking, é necessário definir como será feito teste para determinar se houve erro, uma técnica muito utilizada é a paridade, na qual o numero de (1s) da palavra enviada deve ser PAR para a paridade PAR, e IMPAR para a paridade IMPAR.

Padrão de Comunicação Serial RS232

O RS232 é o padrão serial mais conhecido, a conexão entre os dispositivos é feita com um plug de 25 pinos, mostrado na figura 3.14, nem todos os pinos são utilizados em todas as aplicações, a configuração mais utilizada é da seguinte forma:

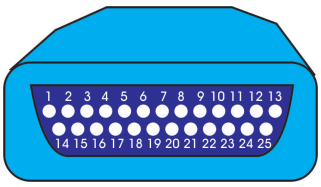

Figura 3.14: Plug Serial

- Pino 1: Ground.
- Pino 2: Serial transmitted data (output).

- Pino 3: Serial received data (input).
- Pino 4: Request to send.
- Pino 5: Clear to send.
- Pino 6: Data set ready.
- Pino 7: Signal ground which acts as a common signal return path.
- Pino 20: Data terminal ready.

Os sinais enviados pelos pinos 5, 6, 7 e 20 são usados para checar se o receptor está pronto para receber os dados, se o emissor está pronto para enviar os dados e se os dados estão prontos para serem enviados. No padrão RS232 o bit 1 é representado por uma tensão entre -5 e -25 geralmente -12, enquanto o bit 0 é representado por uma tensão entre +5 e +25 normalmente +12. O termo "baud rate"é usado para descrever a taxa na qual os bits são enviados, geralmente em bits por segundo, em uma transmissão nem todos os bits enviados são dados propriamente ditos, alguns bits são acrescentados à mensagem para indicar inicio e fim de transmissão, outros bits são acrescentados, e utilizados para checar se houve erro na transmissão, como é o caso do bit de paridade. Um exemplo de transmissão é mostrado na figura 3.15, nessa transmissão a paridade é IMPAR e o dado transmitido é 1101000.

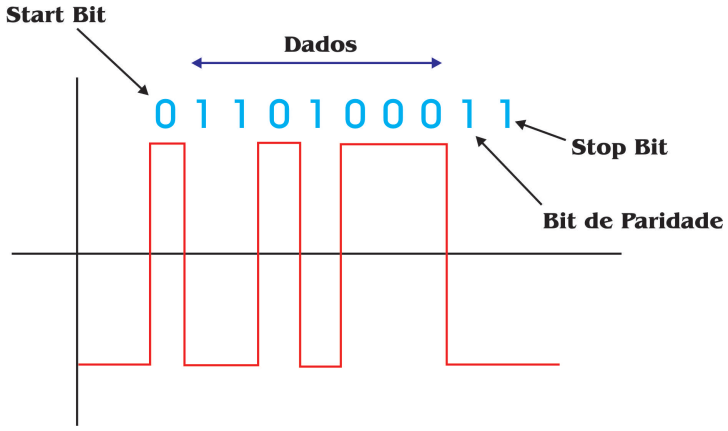

Figura 3.15: Transmissão Serial RS232

Outros protocolos seriais largamente são RS422 e RS423 similares ao RS232, mas são mais eficazes quando a comunicação ocorre em distâncias maiores que 15 metros. A comunicação RS422 usada para longas distâncias

utiliza duas linhas de transmissão com tensões diferentes, quando ruídos afetam as duas linhas igualmente os dados não são afetados.

Uma alternativa para comunicações seriais de longas distâncias é utilizar o sistema mostrado na figura 3.16. O sistema consiste de um circuito, contendo uma fonte de tensão e uma chave, a transmissão dos dados é feita chaveando o circuito, quando a corrente na linha é 20mA o bit enviado é 1, quando a corrente é 0mA o bit enviado e 0.

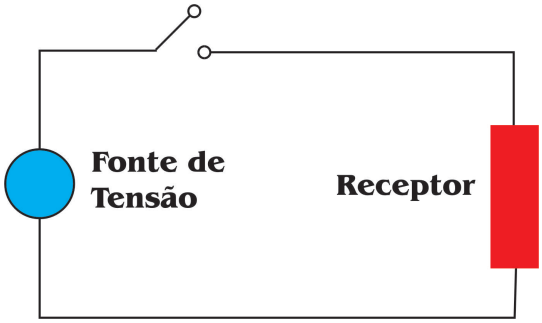

Figura 3.16: Comunicação Serial por Chaveamento

3.10.2 Padrões de Comunicação Paralela

O padrão mais utilizado para comunicação paralela é o IEEE-488. Que foi originalmente desenvolvido pela Hewlett Packard Instrumentation Bus. Esse barramento permitia a comunicação entre dispositivos chamados, "talkers, linsteners and controllers". Linsteners são dispositivos que recebem dados, Talkers dispositivos que enviam dados e Controllers gerenciam o fluxo de dados. O barramento era composto por 24 linhas, 8 bidirecionais para dados e comandos, 5 usados para controle e sinais de status, três para handshaking e 8 para terra. O handshaking é um processo que se inicia quando um dispositivo envia uma mensagem para outro dispositivo indicando que ele quer estabelecer um canal de comunicação. mensagens então são enviadas entre eles a fim de se estabelecer a comunicação. O esquema desse barramento com os dispositivos é mostrado na figura 3.17.

Os comandos enviados pelos controladores são caracterizados por apresentarem a via ATN em 0, caso contrário, ATN for 1, isso indica que há dados nas vias de dados. Os comandos são enviados para os dispositivos de forma individual através de um endereço único no barramento. O endereço então deve ser enviado antes de enviar os dados, através das vias de dados

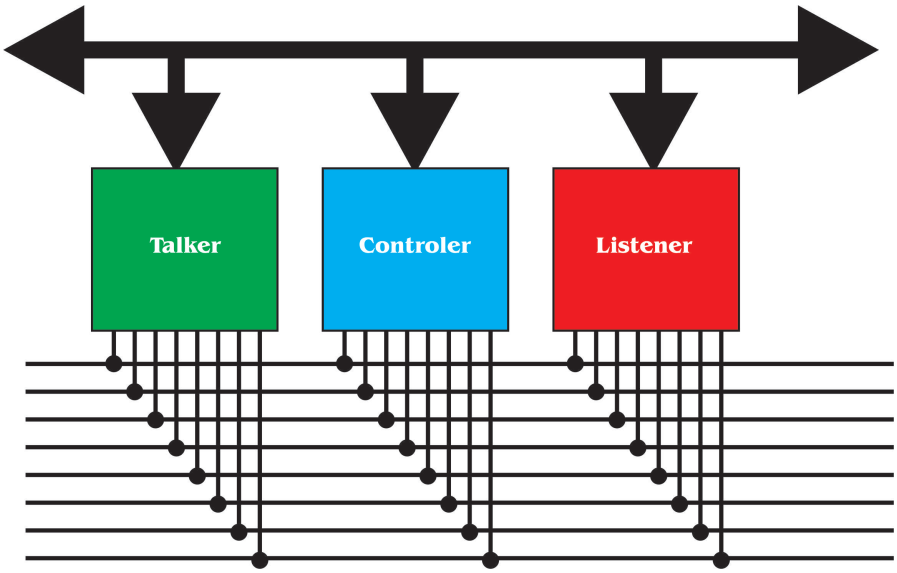

Figura 3.17: Padrão IEEE-488

na forma de uma palavra de 7 bits. Os 5 bits menos significativos fornecem o endereço do dispositivo os demais são bits de controle.

Esses bits de controle funcionam da seguinte forma:

- bit7=0, bit6=0: Comandos enviados para todos os endereços.
- bit7=0, bit6=1: O dispositivo endereçado é setado para ser listener.
- bit7=1, bit6=0: O dispositivo endereçado é setado para ser talker.

É mostrado abaixo de forma simplificada o que significa cada via do barramento IEEE-488:

1. Pin 1 DIO1 Data input/output bit.
2. Pin 2 DIO2 Data input/output bit.
3. Pin 3 DIO3 Data input/output bit.
4. Pin 4 DIO4 Data input/output bit.
5. Pin 5 EOI End-or-identify.
6. Pin 6 DAV Data valid.
7. Pin 7 NRFD Not ready for data.

8. Pin 8 NDAC Not data accepted.
9. Pin 9 IFC Interface clear.
10. Pin 10 SRQ Service request.
11. Pin 11 ATN Attention.
12. Pin 12 SHIELD
13. Pin 13 DIO5 Data input/output bit.
14. Pin 14 DIO6 Data input/output bit.
15. Pin 15 DIO7 Data input/output bit.
16. Pin 16 DIO8 Data input/output bit.
17. Pin 17 REN Remote enable.
18. Pin 18 GND.
19. Pin 19 GND.
20. Pin 20 GND.
21. Pin 21 GND.
22. Pin 22 GND.
23. Pin 23 GND.
24. Pin 24 Logic ground

3.10.3 Protocolos

Para que se possa controlar uma comunicação entre dois dispositivos, é necessário definir onde inicia e termina a comunicação e outros parâmetros, esse conjunto de definições é chamado de Protocolo. Em uma comunicação um dispositivo precisa informar ao outro que está pronto para enviar os dados, assim como o outro deve informar que está pronto para receber.

Uma alternativa para informar os estados dos dispositivos é enviar caracteres adicionais, por exemplo utilizando o protocolo ENQ/ACK dados são enviados para um receptor com um caractere ENQ, quando esse caractere é recebido indica o final do envio de um pacote de dados, nesse momento

o receptor envia ao emissor um caractere ACK, informando assim que está pronto para receber um novo dado.

Uma forma de checar se ocorreram erros nos dados transmitidos é utilizar o bit de paridade, por exemplo, se a paridade é definida como sendo IMPAR, e um bloco de dados a ser enviado é formado pelos seguintes bits, 1011001011, esse bloco tem uma quantidade PAR de "1", logo deve ser adicionado mais um "1"fazendo então o total de "1"IMPAR. Quando o receptor receber o bloco ele fará a soma do número de "1"caso der uma soma PAR o bloco está corrompido e ele pode ou informar que o dado está corrompido ou requisitar ao emissor o re-envio do pacote.

O método do bit de paridade pode detectar quando o bloco está corrompido pela troca de um "1"por um "0"ou vice versa, mas não detecta caso ocorram dois erros desse tipo, uma alternativa para contornar isso é arranjar vários blocos enviados em linhas e fazer a paridade por linhas e colunas, nesse caso o erro só será detectado a cada conjunto de blocos enviados, a figura 3.18 mostra um exemplo dessa técnica utilizando paridade PAR.

Figura 3.18: Utilização do Bit Paridade

3.10.4 Código ASCII

O código mais utilizado para a comunicação utilizando caracteres alfanuméricos é o Código ASCII, esse código é formado por 7bits, sendo capaz de representar todas as letras do alfabeto, minúsculas, maiúsculas e caracteres de pontuação. É ainda possível formar códigos para controle, como por exemplo SOH é o primeiro caractere do cabeçalho de uma comunicação representado por 0000001, STX para começar o texto representado por 0000010 e EOT para fim de transmissão representado por 000 0011.

3.11 Redes

As redes são utilizadas para ligar vários dispositivos uns aos outros, as informações entre eles são enviadas e recebidas respeitando padrões e protocolos previamente estabelecidos. Redes são construídas basicamente de três formas:

Forma de Estrelar: Nesse tipo de rede temos um dispositivo que é o host e os demais são terminais, ou um mestre e os demais escravos, as comunicações entre terminais são intermediadas pelo host este deve ter capacidade de processamento para permitir essas comunicações. Um esquema da rede em anel é mostrado na figura 3.19a.

Barramento: Nesse tipo de rede temos um barramento ao qual todos os terminais são ligados, dessa forma é necessário estabelecer protocolos que evitem que dois terminais estejam escrevendo ao mesmo tempo no barramento. Um esquema da rede em anel é mostrado na figura 3.19b.

Anel: Esse tipo de rede é formada pela ligação de um teminal ao outro fechando um circulo, nesse tipo de comunicação deve-se também criar protocolos de comunicação para evitar que mensagens enviadas por terminais diferentes se misturem. Os métodos Anel e Barramento são conhecidos por peer to peer, nesse tipo de rede cada terminal tem a mesma prioridade. Um esquema da rede em anel é mostrado na figura 3.19c.

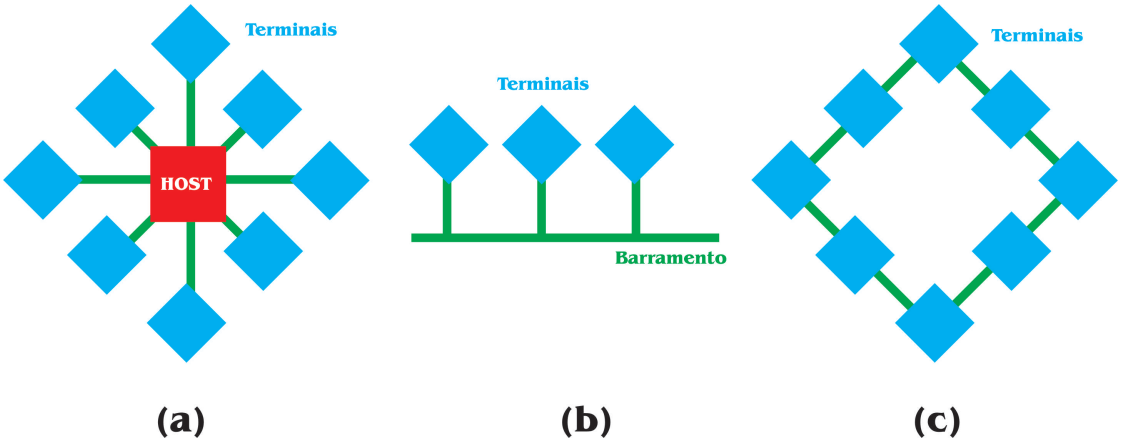

Figura 3.19: Redes: (a) Estrela, (b)Barramento, (c)Anel

Redes em Anel necessitam de um método para evitar que dois terminais iniciem uma transmissão ao mesmo tempo, um dos métodos é o TOKEN, trata-se de um bit que fica circulando entre os terminais da rede, o terminal que desejar transmitir não pode transmitir dados no barramento enquanto

ele não estiver em posse do TOKEN, quando ele enviar o dado o TOKEN vai junto ao dado, após o receptor receber o dado ele libera o TOKEN para que outro terminal ou o mesmo o utilize.

Redes em barramento também devem evitar que dois terminais usem o terminal para transmissão ao mesmo tempo, o método usado é fazer os terminais "olharem"no barramento se está havendo alguma transmissão, caso não, o terminal usa o barramento, esse método é conhecido por CSMA (carrier sense multiple access). Ainda assim pode acontecer de dois terminais "olharem"para o terminal ao mesmo tempo e identificando não estar havendo transmissão, iniciar ao mesmo tempo uma transmissão, nesse caso depois de detectar que está havendo erro na transmissão eles param a transmissão e esperam um tempo randômico antes de iniciar uma nova transmissão, esse método é conhecido por CSMA/CD(carrier sense multiple access with collision detection).

Os fabricantes de CLPs geralmente implementam seus próprios métodos de comunicação, por exemplo, a Mitsubish utiliza o protocolo MelsecNET, Allan Bradley esa a Data Highway Plus, e assim por diante.

3.11.1 Sistemas Distribuídos.

Em sistemas distribuídos os CLPs constituem apenas um nível da hierarquia do sistema, os níveis são basicamente divididos da seguinte forma: No primeiro nível temos dispositivos de entrada e saída tais como sensores e motores, os quais são conectados através de interfaces de entrada e saída as quais constituem um segundo nível, o terceiro nível é constituído por pequenos CLPs e computadores ligados através de uma rede a grandes CLPs que constituem um quarto nível, todos esses níveis podem fazer parte de um sistema ainda maior com Mainframes controlando tudo. Um esquema de um controle com hierarquia é mostrado na figura 3.20.

3.11.2 Padrões de Redes

A Conexão de muitos dispositivos pode apresentar alguns problemas de incompatibilidade, quando as taxas de transmissão ou protocolos de comunicação dos dispositivos são diferentes. Para facilitar a comunicação entre diferentes dispositivos a ISO (Internatiaonal Standards Organization) em 1979 planejou um modelo para ser usado na padronização para open systems interconnection (OSI), o modelo denominado de ISO/OSI model. A respresentação desse sistema é mostrado na figura 3.21.

- Camada 1: Physical, Médium Essa camada é responsável por codi-

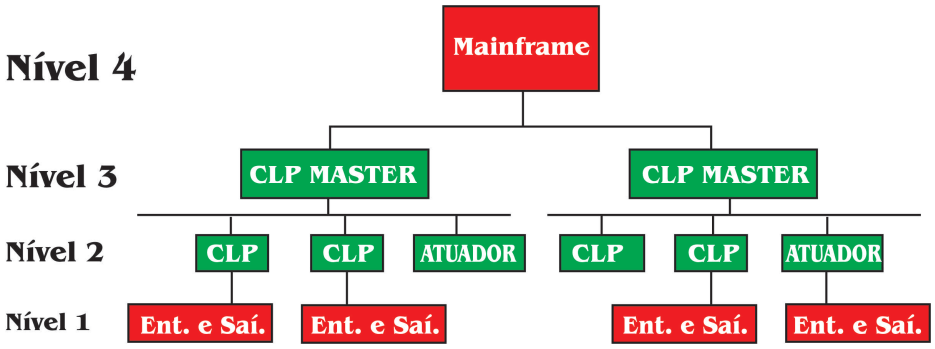

Figura 3.20: Esquema de Sistema Distribuído

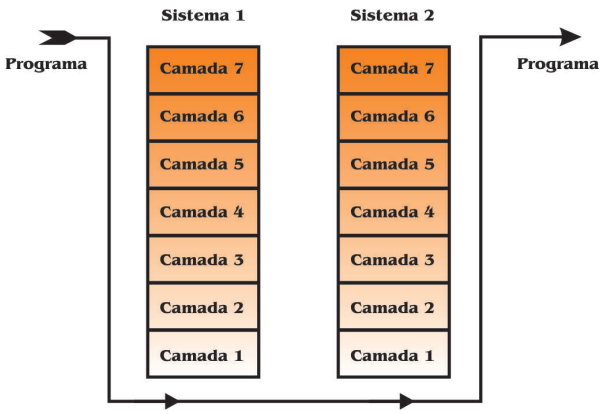

Figura 3.21: Modelo ISO/OSI

ficação e transmissão física da informação, isso inclui sincronismo e transferência dos dados entre os systemas.

- Camada 2: Data link, Define os protocolos para envio e recebimento de informações entre sistemas que estão diretamente conectados um ao outro. Essas funções incluem o controle da seqüência no envio dos dados, e a checagem se houve erro no envio dos dados.

- Camada 3: Network, Define o chaveamento das rotas entre sistemas em uma rede

- Camada 4: Transport, Define os protocolos responsáveis pelo envio de mensagens de uma rede para outra, isso inclui o controle dessa comunicação.

- Camada 5: Session, Esta camada fornece a função para a configuração de usuários em locais separados.

- Camada 6: Presentation, essa sessão assegura que a informação será entregue de forma entendível.

- Camada 7: Application, tem a função de ligar o programa do usuário dentro do processo de comunicação.

Cada camada se comunica somente com as camadas imediatamente abaixo e acima dela. A camada executa suas tarefas e transfere seus resultados para a camada imediatamente abaixo ou acima, isso dá a capacidade de fabricantes desenvolverem dispositivos para trabalhar em uma camada particular, fazendo interface com o hardware de outros fabricantes.

3.12 Processamento das Entradas.

Um CLP está continuamente executando o programa em sua memória, em um loop infinito, cada loop completo é um ciclo. Existem dois modos de processar as entradas, um deles é, modificar as saídas tão logo seja percebido pelo CLP que as entradas mudaram, esse método é chamado de "continuous updating", entretanto ler uma entrada despende um certo tempo, de forma que se muitas entradas são lidas constantemente isso fará com que a execução do programa se torne lenta, uma solução encontrada foi, reservar na memória RAM um bloco específico para entradas e saídas, esses blocos seriam utilizados da seguinte forma: Ao inicio de um ciclo todas as entradas seriam lidas, e armazenadas na memória RAM, as operações são realizadas nos dados relativos as entradas, salvos na memória RAM e os resultados são gravados no bloco destinado as saídas também na memória RAM, ao final do ciclo os dados são transferidos da memória RAM para as saídas atualizando assim o estado dos dispositivos de saída, esse método é chamado de "mass I/O copying"mostrado na figura 3.22

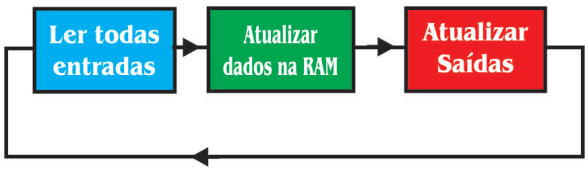

Figura 3.22: Fluxograma: mass I/O copying

Utilizando o método "mass I/O copying"as entradas não são lidas durante todo o tempo, ao invés disso a entrada é lida de forma periódica em intervalos de tempo relativamente curtos, da ordem de 10 a 50ms. Importante observar que uma entrada digital não pode ter uma freqüência maior que a freqüência de leitura das entradas pelo CLP, ou seja se a freqüência de leitura de um CLP for de 50Hz, a freqüência de envio de dados as entradas deve ser preferencialmente menor que 50Hz podendo chegar a ser 50Hz. A freqüência de leitura das entradas depende dos seguintes fatores:

- CPU usada. O tamanho do programa que esta sendo executado.
- O número de entradas e saídas que estão sendo lidas.
- As funções do sistema que estão em uso.

3.13 Endereçamento de Entradas e Saídas

As entradas e saídas de um CLP possuem endereços únicos cada uma delas, em pequenos CLPs esse endereçamento é realizado de forma simplificada, por exemplo o CLP Mitsubishi as entradas são do tipo X400, X401, etc, enquanto as saídas são do tipo Y430, Y431, etc.

Em grandes CLPs, nos quais os módulos de entradas e saídas são encaixados em um rack, como o Allen-Bradley PLC-5, o rack que contem o Processador recebe o número 0, os demais são numerados de forma crescente, 1, 2, 3..., etc. Cada rack contém um número de módulos e cada módulo um número de entradas e/ou saídas. O endereçamento é feito então como mostrado na figura 3.23. Se tivermos uma entrada com o endereço I:012/03 isso significa que ela se encontra no rack 01, no módulo 2 e no terminal 03.

Cada fabricante desenvolve sua própria forma de endereçamento e os manuais vem explicando como fazer esse endereçamento.

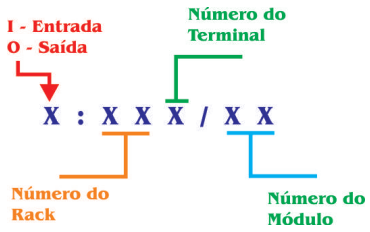

Figura 3.23: Endereçamento de E/S do PCL-5

Capítulo 4

Programando o CLP

4.1 Introdução

Quando desejamos que o CLP execute uma determinada tarefa é necessário escrever e gravar em sua memória as instruções referentes a essa tarefa. A escrita dessas instruções pode ser de várias formas utilizando várias linguagens, no entanto antes de gravar o programa na memória do CLP, o código escrito deve ser convertido em código de máquina, ou seja, uma seqüência do tipo 1101010011011001010... As linguagens de programação podem ser de baixo nível, por exemplo, assembler ou alto nível como C#, ambas requerem do programador um conhecimento relativamente grande da sintaxe da linguagem e lógica de programação.

Os CLPs por outro lado possuem uma linguagem de programação chamada LADDER relativamente simples e que não exige muito conhecimento de programação, algumas horas de estudo dessa linguagem é o suficiente para uma pessoa ser capaz de realizar pequenos projetos de automação, desde que seja de conhecimento dessa pessoa os conceitos estudados nos capítulos anteriores.

LADDER é uma linguagem simbólica e como já dito, de fácil aprendizado, por esse motivo essa linguagem acabou se tornando bastante popular sendo adotada pela maioria dos fabricantes de CLP, em 1993 foi publicado o padrão IEC-1131-3, esse padrão é constituído das seguintes linguagens: LAD, IL, SFC, ST e FBD.

4.2 A programação em LADDER

Para escrever um programa em LADDER usamos símbolos que representam as entradas e as saídas do circuito, bem como funções matemáticas e lógicas, todo o diagrama é desenhado entre duas linhas verticais chamadas

de "power rails"os circuitos e funções são situados em linhas horizontais chamadas de rungs.

Para o melhor entendimento de como funciona os diagramas em LADDER veja a figura 4.1, as setas representam o fluxo da execução do programa.

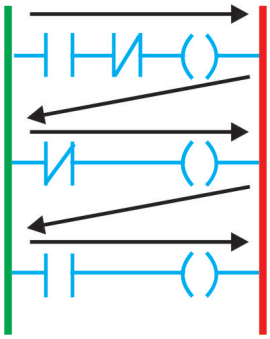

Figura 4.1: Exemplo de um diagrama Ladder

- As linhas verticais dos diagramas representam as "power rails".
- Cada "rung"define uma única operação do processo.
- Um diagrama LADDER é lido da esquerda para direita, de cima para baixo. Assim o primeiro "rung"é lido da esquerda até a direita, então se inicia a leitura do segundo "rung"da esquerda até a direita até que termine o diagrama e então se retorna ao "rung"1, esse processo é chamado de ciclo.
- Cada "rung"é iniciado com uma ou um conjunto de entradas, e termina em uma saída.
- Uma entrada pode ativar várias saídas.
- Um mesmo dispositivo pode aparecer em vários "rungs".
- Entradas e saídas são todas identificadas por seu endereço, esse endereçamento é particular do CLP, e cada fabricante tem seus próprios métodos.

A figura 4.2 mostra alguns símbolos adotados pelo padrão IEC 1131-3, a figura 4.2.a mostra o símbolo utilizado para representar o contato normalmente aberto, a 4.2.b o contato normalmente fechado e a 4.2.c mostra o símbolo utilizado para saída.

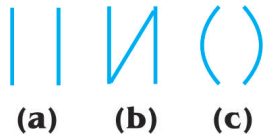

Figura 4.2: Simbologia Ladder

Para exemplificar o uso da linguagem LADDER, suponha que um motor deve ser ativado por um botão (START) e deve parar quando apertado o botão (STOP), quando o motor estiver ligado uma luz deve ser acessa, informando ao usuário que o sistema esta ligado, quando o mesmo estiver parado outra luz deve ser acesa. O diagrama ladder é mostrado na figura 4.3 e um esquema do sistema é mostrado na figura 4.4.

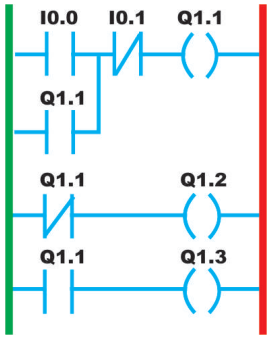

Figura 4.3: Diagrama Ladder

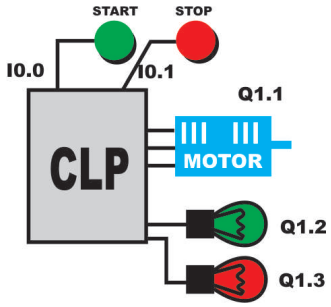

Figura 4.4: Sistema START/STOP para motor DC

O programa funciona da seguinte forma, considere que a saída Q1.1(motor) está desligada, logo a saída Q1.2(lâmpada verde) está ligada, e a saída

Q1.3(lâmpada vermelha esta desligada), a lâmpada verde informa ao usuário que o sistema está parado enquanto a vermelha informa que o sistema está em funcionamento, a entrada I0.0(botão start) é um contato NO, logo quando pressionada ativa a saída Q1.1 que por sua vez ativa a saída Q1.3. Quando se deseja parar o sistema, pressiona-se o botão stop que está ligado a entrada I0.1.

4.3 Funções Lógicas

As entradas deixam passar por elas a energia proveniente do "power rail"esquerdo, para que saídas possam ser ativadas, existem dois tipos de entradas, Normalmente Aberta (NO), deixa passar a energia quando ativada. Normalmente Fechada (NC), deixa passar a energia quando desativada. A figura 4.2 mostra os dois tipos básicos de entradas.

Para realizar funções lógicas (AND, OR, NOR, NAND, XOR), simplesmente podemos combinar várias entradas.

AND. É implementada em ladder colocando-se duas entradas em série, ou seja, apenas quando as duas entradas estão ativas, temos a saída ativada, figura 4.5.

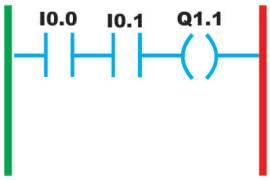

Figura 4.5: AND em LADDER

OR. É implementada colocando-se duas entradas em paralelo, dessa forma se pelo menos uma das entradas estiver ativa, a saída estará ativada, figura 4.6.

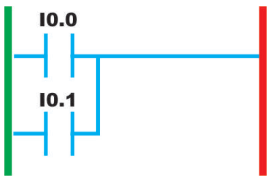

Figura 4.6: OR em LADDER

NOR. Implementada através da OR, colocando-se em sua saída um NC, ou da mesma forma que a AND, mas utilizando NCs, figura 4.7.

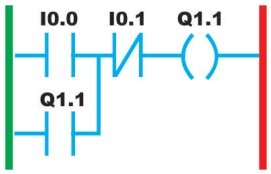

Figura 4.7: NOR em LADDER

LATCH. É possível utilizar como entrada um sinal da saída, dessa forma podemos construir funções que possuem realimentação, como é o caso da latch, essa função guarda um resultado na saída mesmo que a entrada seja cessada, figura 4.8.

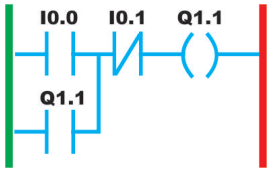

Figura 4.8: LATCH em LADDER

4.4 Escrevendo e carregando programas

Para escrever o programa e coloca-lo no CLP é necessário um outro hardware com um software específico, este hardware pode ser um PC utilizando o software STEP-7, os códigos são escritos em LADDER ou outra linguagem e então são traduzidos para a linguagem de máquina, só então são carregados na memória do CLP.

4.5 Blocos de Função

São blocos utilizados para representar funções, tomemos como exemplo uma função AND, cujas entradas são I0.0 e I0.1 e saída seja Q1.0, o bloco dessa função é representado na figura 4.9.a. Note que deve ser colocado um símbolo no bloco identificando sua função. A figura 4.9.b mostra o bloco da função XOR.

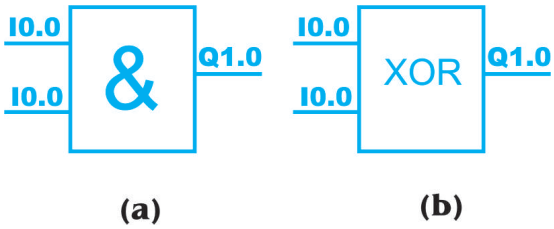

Figura 4.9: Blocos de Função

A figura 4.10 mostra como ficaria o bloco da função latch. Cujas entradas sejam A e B e a saída seja Q.

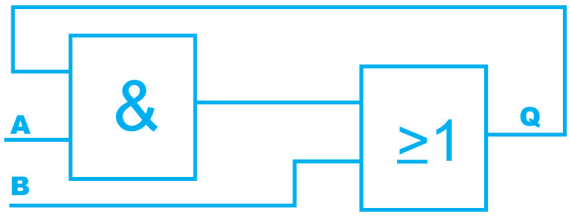

Figura 4.10: Blocos da função LATCH

4.6 Algebra de Boole

A álgebra booleana foi criada pelo matemático inglês George Boole (1815-1864). Boole construiu sua lógica a partir de símbolos, representando as expressões por letras e ligando-as através de conectivos - símbolos algébricos. A álgebra booleana trabalha com apenas duas grandezas: falso ou verdadeiro. Atualmente, todos os sistemas digitais são baseados nela, relacionando os níveis lógicos 0 (falso) e 1 (verdadeiro) com a passagem ou ausência de corrente elétrica. O aprofundamento desse assunto foge ao propósito desse material,cabendo aos leitores buscarem materiais específicos sobre a Algebra de Boole a fim de entendê-la e dominá-la.

4.7 Outras Linguagens de Programação

Outras linguagens de programação podem ser utilizadas para programar o CLP

4.7.1 IL-Instruction List

trata-se uma linguagem de programação composta por uma lista de instruções em forma de mnemônicos, possui as seguintes características:

- Mnemônicos representam operações lógicas booleanas e comandos de transferência de dados.
- Originalmente cada fabricante oferecia seu próprio conjunto de mnemônicos.
- Mnemônicos unificados pela norma IEC 1131-3.

Instruções

Repertório básico de comandos:

- E: operação lógica "E"entre acumulador e operando;
- EN: operação lógica "E"entre acumulador e negação do operando;
- OU: operação lógica "OU"entre acumulador e operando;
- OUN: operação lógica "OU"entre acumulador e a negação do operando;
- CAR: carrega conteúdo do operando no acumulador;
- CARN: carrega conteúdo negado do operando no acumulador; o Repertório básico de comandos (cont):
- S: seta operando em 1, se o conteúdo do acumulador for 1;
- R: reseta operando em 0, se o conteúdo do acumulador for 1;
- ARM: armazena conteúdo do acumulador no operando indicado;
- ARMN: armazena conteúdo negado do acumulador no operando indicado;
- TEM: executa um retardo com temporização definida por "valor".

A linguagem contém ainda mnemônicos para as seguintes operações:

- deslocamentos (shifts) à esquerda e à direita;
- movimentadores: movimentam entrada para memória, memória para saída, etc;

- Op. aritméticas: permitem somar, subtrair, multiplicar e dividir operandos;
- Op. binárias: operações E, OU e OU exclusivo entre operandos;
- contadores: operações de contagem simples (incremental), contagem bidirecional (incremental / decremental) e temporização (delay);
- conversão: operações de conversão A/D, D/A, Binário / Decimal e Decimal / Binário;
- comunicação: envio e recepção de mensagens via rede;

Como exemplo da utilização da linguagem de programação IL temos o seguinte programa:

Problema

S1=(X2.X3+X4).X1 S2=S3=S1.X5

Programa do CLP:

```
CAR x2 ; coloca x2 no acumulador
E x3 ; acum = acum AND x3
OUN x4 ; acum = acum OR NOT x4
E x1 ; acum = acum AND x1
ARM s1 ; armazena acum em s1
EN x5 ; acum = acum AND NOT x5
ARM s2 ; armazena acum em s2
ARM s3 ; armazena acum em s3
```

Uma comparação entre LADDER e IL é mostrada na figura 4.11.

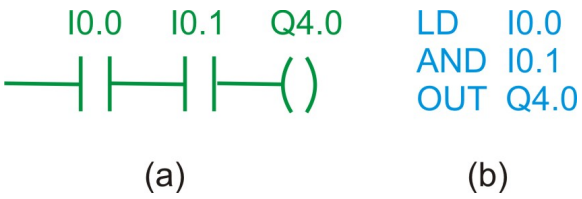

Figura 4.11: (a) LADDER, (b) IL

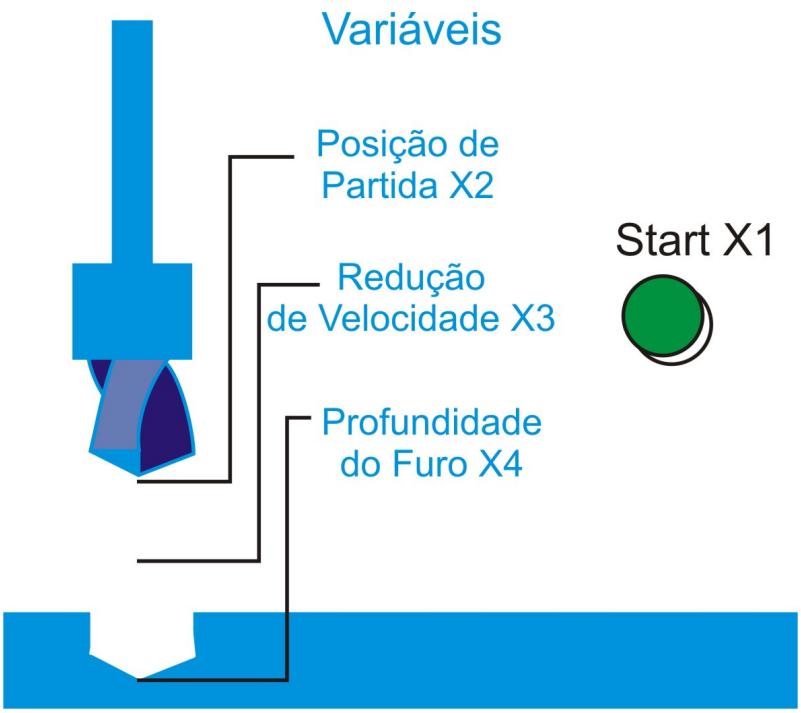

Figura 4.12: Aplicação linguagem IL

Aplicação

Para ilustrar o uso da linguagem IL, vamos mostrar o seguinte exemplo prático, esquematizado na figura 4.12

Sinais de entrada e saída

Sinais de Entrada:

- x1: comando de partida (operador);
- x2: furadeira na posição de partida;
- x3: ponto de redução alcançado;
- x4: profundidade de perfuração alcançada.

Sinais de Saída:

- y1: ligar/desligar avanço rápido;
- y2: ligar/desligar avanço lento com rotação;

- y3: ligar/desligar retrocesso rápido.

Operação básica

Se a furadeira estiver na posição de partida (x2=1) e um comando de partida foi dado (x1=1), ligar avanço rápido (y1=1) até atingir o ponto de redução (x3=1). o Neste ponto, desligar o avanço rápido (y1=0) e ligar o avanço lento com rotação (y2=1). o Ao atingir a profundidade de perfuração desejada (x4=1), desligar o avanço lento com rotação (y2=0) e ligar o retrocesso rápido (y3=1), até retornar a posição de partida

Programa CLP

```
CAR x1 ; SE comando de partida
E x2 ; E posição de partida
S y1 ; ENTÃO liga avanço rápido
CAR y1 ; SE avanço rápido ligado
E x3 ; E ponto de redução
R y1 ; ENTÃO desl. avanço rápido
S y2 ; e liga avanço lento com rotação
CAR y2 ; SE avanço lento com rotação ligado
E x4 ; E profundidade de perfuração atingida
R y2 ; ENTÃO desl. avanço lento com rotação
S y3 ; e liga retrocesso rápido
CAR y3 ; SE retrocesso rápido ligado
E x2 ; E pos. partida atingida
R y3 ; desliga retrocesso rápido
```

O leitor pode ter chegado a conclusão que programar em Ladder é muito mais simples do que programar em IL, o que é verdade e esse fato faz com que Ladder seja largamente usada, algumas vantagens e desvantagens da linguagem IL são listadas abaixo:

Vantagens

- Correspondência entre comandos da linguagem e as instruções assembly do processador, facilitando uma estimativa do tempo de execução do programa;

- Documentação mais compacta do que a equivalente com relês.

Desvantagens

- Necessidade de familiarização do operador com álgebra booleana;
- Necessidade de uma certa familiarização com programação em assembly;
- Alterações trabalhosas nas lógicas já implementadas.

Existem ainda outras linguagens como ST (structured text) e SFC (sequencial function chart).

4.7.2 SFC

O SFC descreve graficamente o comportamento seqüencial de um programa de controle. É derivado das redes de Petri e da norma IEC 848 Grafcet, com as alterações necessárias para converter a representação de uma documentação padrão para um conjunto de elementos de controle de execução. O SFC estrutura a organização interna do programa e ajuda a decompor o problema de controle em partes gerenciáveis, enquanto mantém a sua visão geral. O SFC consiste de Passos, interligados com blocos de Ações e Transições. Cada passo representa um estado particular do sistema sendo controlado. Uma transição é associada com uma condição, a qual, quando verdadeira, causa a desativação do passo anterior à mesma e a ativação do passo seguinte. Passos são ligados com blocos de ações, desempenhando uma determinada ação de controle. Devido a sua estrutura geral, o SFC funciona também como uma ferramenta de comunicação, integrando pessoas de diferentes formações, departamentos e países.

Caso seja desejado descrever em SFC um semáforo nós teríamos o programa mostrado na figura 4.13. As condições 1 e 2 são simplesmente assegurar que tenhamos uma espera de 1 minuto antes do próximo estado ser alcançado.

4.7.3 ST

Texto Estruturado é uma linguagem de alto nível muito poderosa, com raízes em Ada, Pascal e "C". Contém todos os elementos essenciais de uma linguagem de programação moderna, incluindo condicionais (IF-THEN-ELSE e CASE OF) e iterações (FOR, WHILE e REPEAT). Estes elementos também podem ser aninhados. Esta linguagem é excelente para a definição de blocos funcionais complexos.

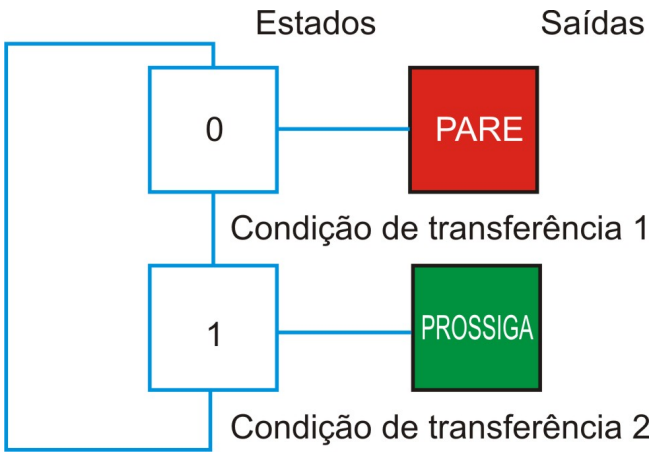

Figura 4.13: Exemplo linguagem SFC

Exemplo em ST

```
I:=25;
WHILE J<5 DO
    Z:= F(I+J);
END_WHILE

IF B_1 THEN
    %QW100:= INT_TO_BCD(Display)
ENDIF

CASE    TW OF
    1,5:    TEMP := TEMP_1;
    2:  TEMP := 40;
    4:  TEMP := FTMP(TEMP_2);
ELSE
    TEMP := 0;
    B_ERROR :=1;
END_CASE
```

Operadores de condição IF, THEN e ELSE, são usados para ativar uma certa ação quando uma certa condição é alcançada. Por exemplo

```
IF(condição) THEN Ação 1; Ação 2; Ação 3; . . . Ação 4; ELSE Ação 5;
Ação 6; End_IF;
```

Pelo motivo de ST ser uma linguagem bem mais complexa que as demais mostradas nesse material, é recomendável que o leitor procure estudá-la em um material mais específico.

Capítulo 5

Dispositivos Internos dos CLPs

5.1 Relés Internos

Vamos tratar agora de um elemento interno dos CLPs, os Relés Internos. Esses elementos são utilizados durante a programação do CLP para fornecer funções especiais, na verdade os Relés Internos são bits na memória que podem apresentar dois estados ON e OFF, esses bits podem ser utilizados para chavear dispositivos externos daí o nome Relé Interno. Um pequeno CLP tem cerca de cem relés internos, alguns deles são alimentados por baterias, para que não percam seus dados após uma queda de tensão.

O esquema de um relé interno é mostrado na figura 5.1, nesse esquema está sendo utilizado um Relé Interno com contato normalmente aberto, quando o contato é fechado o relé fecha e então ativa o dispositivo de saída.

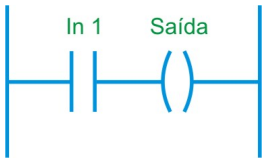

Figura 5.1: Relé Interno

Relés Internos podem ser combinados para a construção de funções mais complexas, para distinguir entre Saídas para Relés Internos e Saídas para Relés Externos, os fabricantes usam diferentes termos e formas de endereçamento para os relés internos, por exemplo, A Mitsubishi utiliza M11, M101, etc. A Siemens utiliza F0.0, F0.1, etc.

5.1.1 Programas em Ladder

Em programas Ladder para representar uma saída usamos o símbolo mostrado na figura 5.2.a onde XX representa o endereço do dispositivo de saída, para representar uma entrada utilizamos o símbolo 5.2.b onde XX representa o endereço do dispositivo de entrada. No caso de usarmos um CLP Mitsubishi e quisermos endereçar a saída e a entrada para um relé interno de endereço M100, colocamos M100 no lugar de XX.

Figura 5.2: Saída e Entrada em Ladder

Programas com múltiplas entradas

Para a construção de funções mais complexas podemos combinar vários dispositivos de entrada, como mostrado na figura 5.3, nessa figura temos a seguinte expressão:

$$RI1 = ((In1(OR)In2)(AND)(In3))$$

Essa função nos diz que: Quando as entradas

$$In1(OR)In2$$

forem ativadas, "E"a entrada In3 também estiver ativada, o relé interno RI1 deve ser ativado. passada a primeira etapa do programa, o resultado RI 1 é utilizado em conjunto com a entrada In4, da seguinte forma:

$$Out1 = (RI1(AND)In4)$$

Quando a entrada In4 for ativada e RI1 estiver ativa, a saída Out1 deve ser ativada.

Programas com circuitos Latch

Um exemplo de um programa latch é mostrado na figura 5.4, nesse programa temos o seguinte comportamento, suponha que o relé interno RI 1 tem um estado inicial desativado, quando a entrada In 1 for ativada a saída Out 1 será ativada, para resetar a saída, ativamos a entrada In 2, que ativará o RI 1, e esse desativará a saída Out 1.

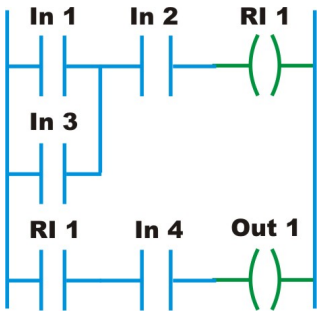

Figura 5.3: Programa em Ladder com Multiplas Entradas

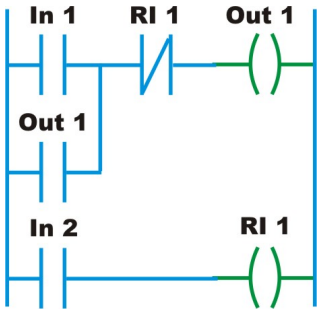

Figura 5.4: Célula Latch em Ladder

Operação One-Shot

As vezes é necessário desenvolver funções que ficam ativas apenas durante um ciclo, alguns CLPs vem com essas funções já construídas em blocos, no entanto não é difícil construí-las, é mostrado na figura 5.5 como realizar esse tipo de função, considere que CC está desativado, note que quando a entrada In 1 é ativada a saída Out 1 é ativada, e logo após CC é ativada, mas como CC só é ativada depois que Out 1 é ativada, ela só terá efeito em Out 1 no próximo ciclo.

Funções Set e Reset

Uma função que é bastante utilizada é a de set ou reset, Os símbolos dessas duas funções são mostrados na figura 5.6, note que quando o coil é resetado ele só voltará a ser ativado quando for utilizada a função set e vice-versa.

Podemos combinar os símbolos de Set e Reset em um único símbolo, dessa forma temos o símbolo mostrado na figura 5.7, esse símbolo é chamado de SR

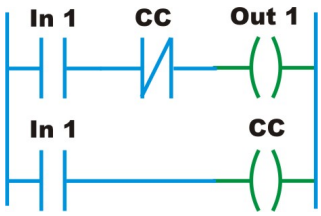

Figura 5.5: Operação One-Shot em Ladder

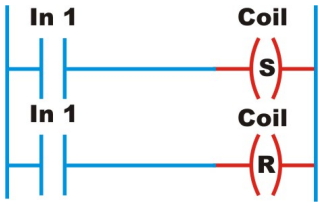

Figura 5.6: Set e Reset em Ladder

Memory, um problema que poderia aparecer é, quando as duas entradas são ativadas o que acontece? Bem, nesse caso existe uma ordem de prioridades, que geralmente o Reset tem maior prioridade. Portanto se ambos forem ativados ao mesmo tempo a saída é resetada. Esse mesmo bloco pode ser utilizado para implementar a função Flip-Flop.

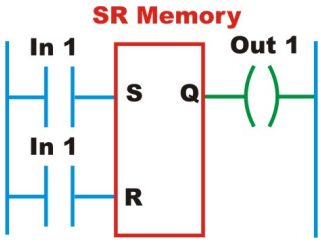

Figura 5.7: SR-Memory ou Flip-Flop

Relé de Controle Mestre

Quando precisamos controlar o acionamento de uma grande quantidade de dispositivos de entrada e saída, podemos utilizar um relé de controle mestre, a figura 5.8 mostra como deve ser programado esse tipo de controle, quando In 1 está desativada então MC 1 está desativado e portanto tudo

que estiver entre MC 1 e MCR 1 está desativado, desse forma caso In 1 esteja desativada, mesmo que sejam ativadas as entradas In 2 e In 3 as suas respectivas saídas continuarão desativadas.

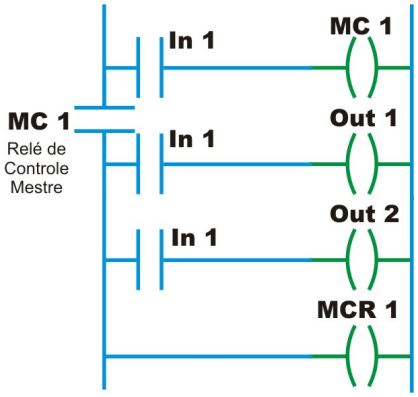

Figura 5.8: Relé de Controle Mestre

Exemplo de Programa

Como primeiro exemplo podemos mostrar um sistema para o sensoriamento de incêndios, nesse caso pode-se utilizar a função SR Memory, tendo nas entradas um conjunto de sensores e um botão para parar o alarme, e na saída podemos colocar a sirene que emite o alarme, o sistema é esquematizado e mostrado na figura 5.9.

Note que os sensores são combinados de um modo que formam uma função OR, sendo que qualquer um deles que detecte a ocorrência de fogo ira enviar ao bloco SR o sinal que fará ele setar a saída, soando então o alarme, nesse caso temos ligada a entrada Reset um botão que quando pressionado reseta a saída, note que aqui temos prioridade do reset.

5.2 Desvios e Chamadas

Desvios e chamadas são utilizados para alterar o fluxo normal de um programa, e para minimizar o código do mesmo, quando necessitamos repetir várias vezes um processo em um programa, podemos escrever esse processo apenas uma vez, e chama-lo repetidas vezes durante a execução do programa completo, isso nos da a possibilidade de reutilizar funções em vários projetos.

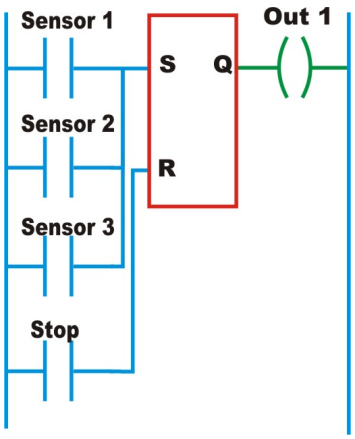

Figura 5.9: Detector de Incêndios

Desvios

Uma função de desvio condicional que está disponível em quase todos os CLPs é a seguinte:

```
IF(condições) THEN Instruções a serem executadas caso a condição
seja satisfeita. ELSE Instruções a serem executadas caso a condição
não seja satisfeita.
```

O exemplo do programa mostrado na figura 5.9, poderia ser modificado para que, apenas quando a temperatura for maior que um valor pré-fixado o sistema soaria o alarme.

A figura 5.10 esquematiza melhor que acontece quando ocorre um desvio, veja que, se In1 estiver ativo, então a função Jump é ativada e o fluxo do programa é alterado de modo que, os rungs 2 e 3 não são exeutados, o programa pula para o rung 4 e daí continua sua execução normalmente, caso In1 esteja desativada então os rungs 2 e 3 são executados normalmente.

Com programas para CLPs Siemens os desvios condicionais são representados como mostra a figura 5.11, no caso JMP é executada sempre que tem como entrada 1, enquanto JMPN é executada sempre que sua entrada é 0, o destino de ambas as instruções é apontado pelo símbolo DEST.

É possível termos desvios em cascata, o esquema é mostrado na figura 5.12, temos os seguintes casos, quando In1 e In2 são 0, então o programa é executado por completo, se In1 é 1, o programa é desviado do rung1 para o rung8, não executando portanto os rungs intermediários, independente do

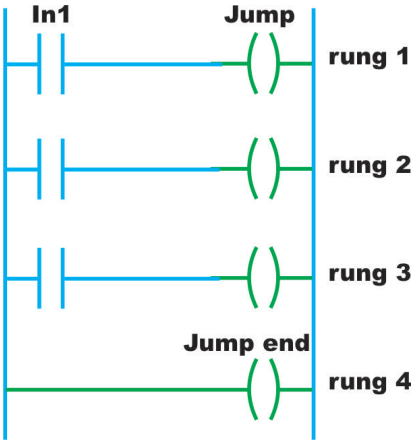

Figura 5.10: Desvios Condicionais

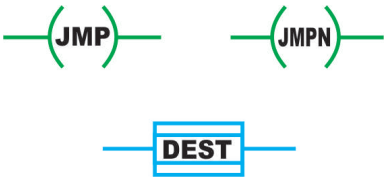

Figura 5.11: Desvios Condicionais/Siemens

In2 ser 0 ou 1. Se In1 é 0 e In2 é 1 então o programa será executado normalmente até chegar ao rung 3, daí ele é desviado para o rung 6 e prossegue normalmente até o final.

Subrotinas

Pequenos programas dentro de programas maiores podem ser classificados como subrotinas, utilizar subrotinas é uma técnica que visa simplificar a resolução de problemas menores dividindo-os em várias partes, cada parte sendo executada por um programa menor, isso nos possibilita utilizar essas mesmas subrotinas em outros programas.

Existem diferentes formas de chamar uma subrotina e isso depende do fabricante, por exemplo, para os CLPs Siemens usa-se a função CALL para chamar uma subrotina e a função RET para retornar dela para o programa principal, Podemos ainda usar uma função que vem em um bloco, esse bloco é mostrado na figura 5.13, note que se EN é conectado diretamente ao power rail a subrotina sempre será chamada, podemos entretanto colocar qualquer

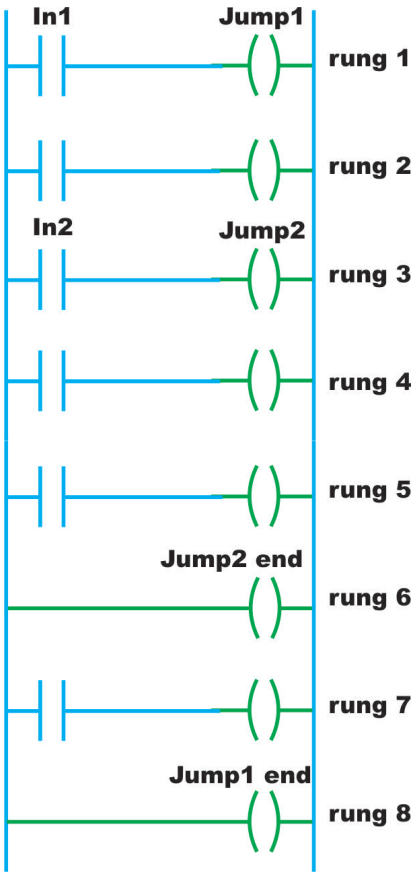

Figura 5.12: Desvios em cascata

condição entre o power rail e o EN, de forma que o EN so será ativado se a condição for satisfeita.

É mostrado na figura 5.14 um programa utilizando funções de chamada de subrotinas para os CLPs Mitsubishi.

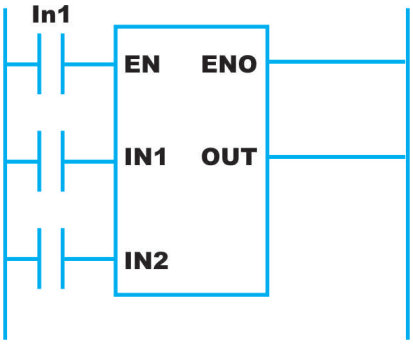

Figura 5.13: Chamada de Subrotinas

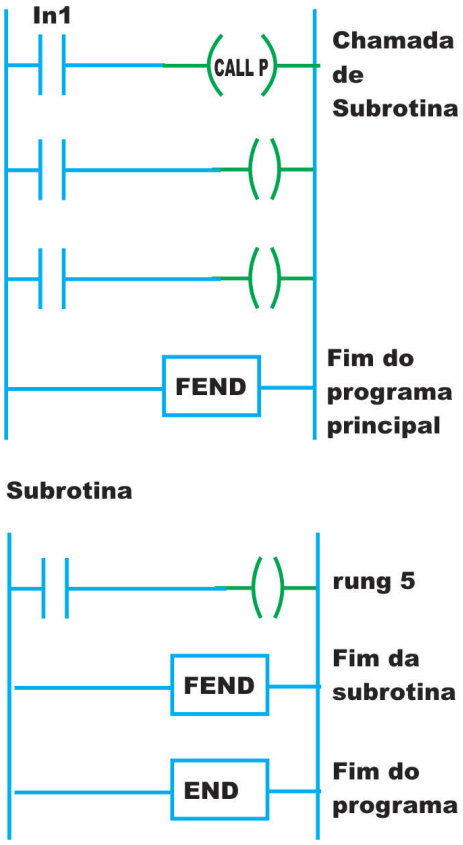

Figura 5.14: Chamada de Subrotinas para CLPs Mitsubshi

5.3 Timers

Em automação industrial muitas tarefas devem ser inicializadas em instantes específicos e devem durar por um período previamente determinado,

tomemos como exemplo uma fabrica de peças de cerâmica, após a cerâmica ser moldada ela deve ir para um forno e receber um aquecimento durante um certo tempo, esse sistema é mostrado na figura 5.15. Sistemas como esse são controlados através de timers, esses contam frações de segundo utilizando o clock da CPU.

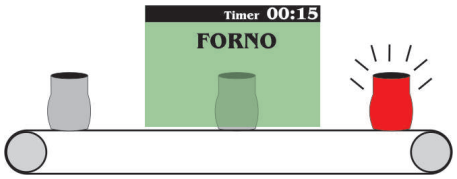

Figura 5.15: Forno para Cerâmicas

5.3.1 Tipos de Timers

Os timers possuem uma entrada (sinal para ativar o timer) e uma saída (resposta do timer), de acordo com a necessidade podemos escolher dentre os diversos tipos de timers que os CLPs fornecem, dentre eles temos os seguintes:

- On-Delay-Timer: Esse tipo de timer ativa sua saída após ter na sua entrada um sinal alto durante um determinado tempo. Sua saída continua ativada enquanto a entrada estiver em sinal alto, a figura 5.16 mostra como se comporta esse timer.

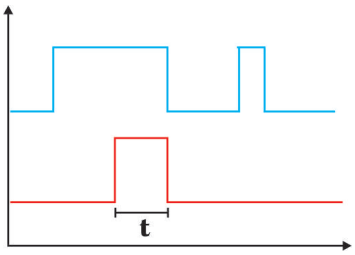

Figura 5.16: On-Delay-Timer

- Off-Delay-Timer: Esse tipo de timer ativa sua saída a partir do momento que sua entrada vai para nível alto, a partir desse momento se o sinal de entrada vai para nível baixo, a saída fica ativada durante um tempo predeterminado, para melhor compreensão pense no sistema de uma lâmpada utilizada para iluminar escadas, a pessoa acende a

lâmpada e ela permanece acesa durante alguns minutos, o ato de acender a lâmpada seria a entrada do timer e a saída seria a tensão na lâmpada, o tempo de ajuste do timer seria o tempo que a luz ficou acesa após o usuário largar o botão, a resposta desse timer é mostrada na figura 5.17.

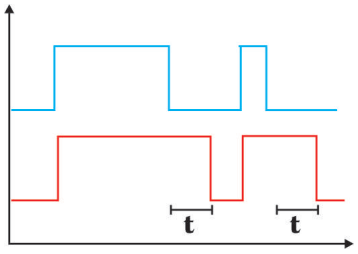

Figura 5.17: Off-Delay-Timer

- Pulse-Timer: Esse tipo de time funciona da seguinte forma, quando sua entrada é vai para nível alto, sua saída produz um pulso de nível alto, no entanto a entrada deve estar em nível alto durante todo o período do pulso de saída, e caso a entrada vá para nível baixo o pulso de saída vai então para nível baixo, caso a entrada permaneça em nível alto durante um tempo muito maior que o período do pulso, a saída irá para nível baixo assim que decorrido o período do pulso, como mostra a figura 5.18.

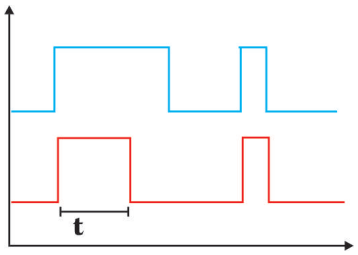

Figura 5.18: Pulse Timer

Os timers possuem um time-base que especifica a precisão desse timer, se por exemplo temos um timer setado com um valor 1000 e seu time-base é de 1ms então, o período desse timer é de 1 segundo.

5.3.2 Programando os Timers

O modo de programar os timers, e quais tipos de timers estão disponíveis em um CLP varia de acordo com o fabricante, no entanto um timer que é encontrado em praticamente todos os CLPs, pequenos ou grandes, é o On-Delay-Timer, A figura 5.19 mostra como seria a programação desse tipo de timer para CLPs Mitsubishi, veja que o Timer T450 K5 tem como sinal de entrada X400, o K5 especifica por quanto deve ser multiplicado o time-base para se conseguir o período. Supondo que o time-base seja de 100ms, se a entrada X400 for ativada durante meio segundo então o timer será ativado, ativando então a saída Y430, observe que a saída estará ativa enquanto a entrada assim estiver.

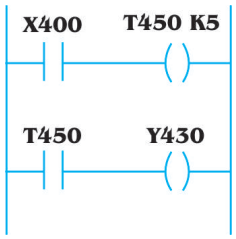

Figura 5.19: ODT-Mitsubishi

A figura 5.20 mostra o mesmo processo realizado para CLPs da Siemens, nesse caso a variável TV do tipo KT é quem especifica o período do timer, para resetar esse timer basta colocar nível alto na entrada R.

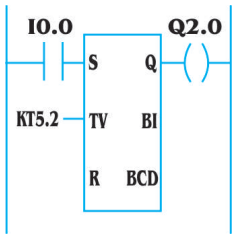

Figura 5.20: ODT-Siemens

5.3.3 Acionamento Seqüenciado

As vezes pode ser necessário controlar um processo no qual vários atuadores são acionados de forma seqüencial respeitando um certo tempo entre o

acionamento dos mesmos, a figura 5.21 mostra como é possível acionar duas saídas em seqüência, com um delay entre elas de 10 segundos.

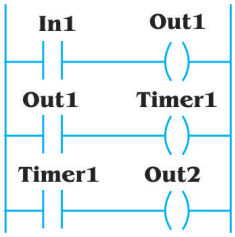

Figura 5.21: Acionamento Seqüenciado

Outro exemplo de acionamento seqüenciado é mostrado na figura 5.22, nesse caso temos um controle START/STOP para um sistema que possui 1 lampada e 2 motores, considerando que RI1 está em 0, quando o START é pressionado então o relé interno RI1 é acionado, juntamente com a lâmpada 1, essa lâmpada tem como objetivo sinalizar ao operador que o sistema está a se preparando para iniciar seu funcionamento, o RI1 serve como entrada para T1 e T2 esses timers possuem períodos diferentes, considerando que T1¡T2 o Motor 1 será então acionado passado algum tempo o Mortor 2 é também acionado, se o operador quiser parar todo o processo basta apertar STOP.

5.3.4 Timers em Cascata

Se por algum motivo um timer não fornecer um período grande o bastante, eles podem ser combinado em cascata para se conseguir praticamente qualquer valor de período, como mostrado na figura 5.23.

5.3.5 PWM

Dois On-Delay-Timers podem ser combinados para a construção de um sinal pwm, no qual a largura de duty cicle pode ser especificada através do período de um dos ODT enquanto o período do pwm seria especificado através da soma dos períodos dos ODTs. A figura 5.24 mostra como ficaria o programa em ladder e como se comportariam as saídas dos timers e do PWM.

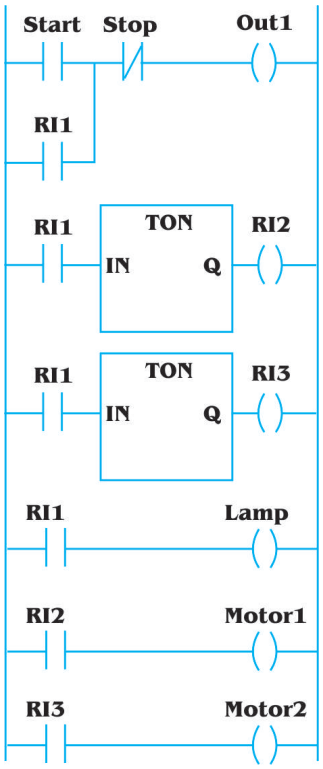

Figura 5.22: Acionamento Seqüenciado

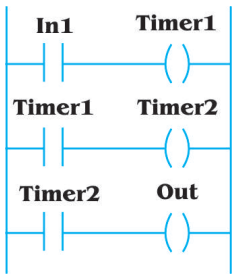

Figura 5.23: Timer em Cascata

5.4 Contadores

Contadores são dispositivos internos dos CLPs utilizados em operações de contagem de processos, como por exemplo o número de peças que passam por uma determinado setor da linha de produção, dentre diversos outros tipos de contagens.

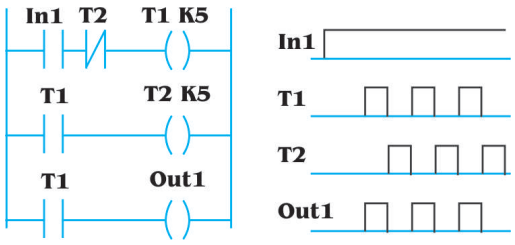

Figura 5.24: PWM

5.4.1 Tipos de Contadores

Existem dois tipos de contadores, crescentes e decrescentes. Os crescentes podem funcionar da seguinte forma, contando a partir de zero até um valor previamente estabelecido ou ainda contando a partir de um valor estabelecido até outro, em ambos os casos toda vez que a entrada do contador é ativada seu valor atual é incrementado até chegar ao valor limite, quando isso acontece sua saída é ativada, e pode ser usada para ativar um relé interno ou ainda uma saída analógica e/ou digital. Os contadores decrescentes funcionam de forma semelhante, uma vez que a sua entrada foi ativada, seu valor é decrementado até que se chegue ao valor zero, daí sua saída é ativada e pode ser usada para fins específicos, como no caso do contador crescente. Dependendo do fabricante o CLP pode vim com ambos os tipos de contador. A figura 5.25 mostra o esquema de um contador e um reset para os CLPs da Siemens.

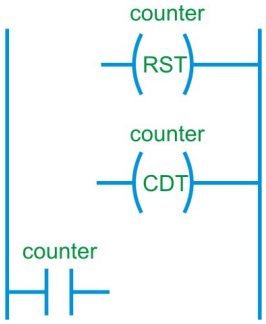

Figura 5.25: Contador

5.4.2 Programando o contador

A figura 5.26 mostra um esquema básico da programação de um contador, suponhamos que o contador é do tipo crescente e inicia sua contagem de zero,

e o valor estabelecido é de 5, ou seja após 5 pulsos terem chegado a entrada In2 então o contador é ativado, ativando dessa forma a Saída, caso um pulso seja enviado a In1 então o contador é resetado.

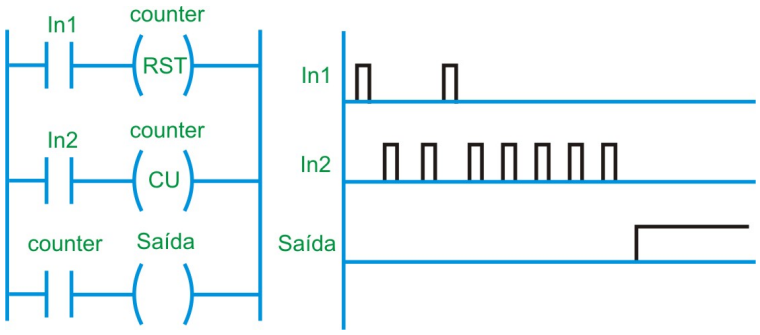

Figura 5.26: Programação Básica de um Contador

A figura 5.27 mostra o programa apresentado na figura 5.26, para CLPs da Siemens. "O CV BCD"é o valor do contador em BCD enquanto que "CV"é o valor em binário, "S"é usado para ativar o contador e incrementa-lo e "R"para resetar.

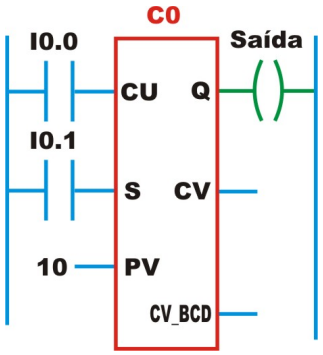

Figura 5.27: Programa para Contador SIEMENS

5.4.3 Contadores de Incremento e Decremento

É possível utilizar contadores de incremento e decremento juntos, por exemplo, uma linha de montagem pode ter um instalado um sensor de passagem de objetos no inicio da linha de montagem e outro no final da linha de montagem, quando um objeto chega a linha de montagem o sensor incrementa o contador, e quando o objeto passa pelo final da linha de montagem

o outro sensor decrementa o contador, dessa forma é possível não só saber quantos objetos estão no momento na linha de montagem, bem como desligar o sistema caso nenhum objeto esteja na linha de montagem, esse programa é mostrado na figura 5.28 para o CLP S7-300 da Siemens.

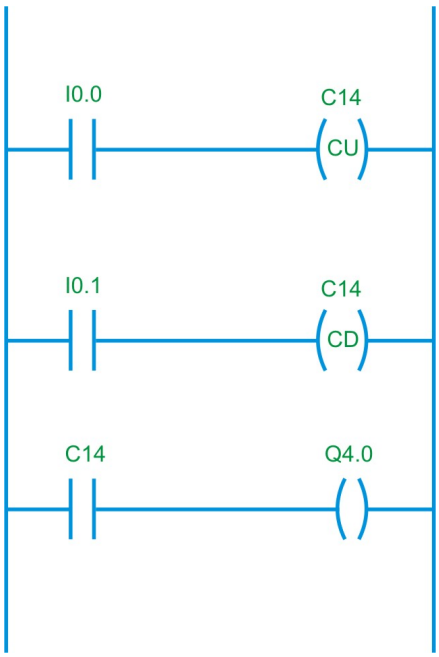

Figura 5.28: Programa para CLP SIEMENS

Contadores de incremento e decremento estão disponíveis em um só, a figura 5.29 mostra o símbolo usado para descrever esses contadores utilizando o padrão IEC 1131-3. O Contador tem duas entradas CU e CD, quando um pulso é detectado em CU o contador incrementa seu valor atual, quando um pulso é detectado em CD o contador decrementa seu valor atua, quando o contador chega a zero a saída QD vai para nível alto e o contador para de decrementar, se o contador chegar no máximo valor PV, o QU vai para nível alto e o contador para de incrementar. CV é o valor atual, R é usado para resetar o contador.

5.4.4 Timers com Contadores

Um típico Timer pode contar até um valor de 16 bits isso nos dá aproximadamente 32.726, se esse relógio tem como time-base 1s então ele pode contar 32.726 segundos, timers podem ser combinados com contadores, e dessa forma pode-se contar longos períodos de tempo, por exemplo, suponha

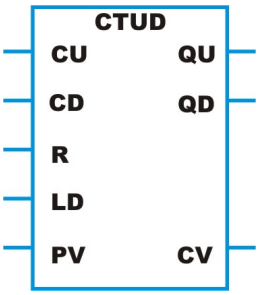

Figura 5.29: Contador Duplo

que tenhamos um timer com time-base 1 segundo, e um valor preset de 3600, mostrado na figura 5.30, esse timer então pode contar 3600 segundos, quando isso acontece DN é setado em 1, DN então incrementa o contador, e reseta o timer, esse processo se repete até que o contador cujo preset está setado em 23 chegue a esse valor, lembrando que o mesmo começa do valor zero.

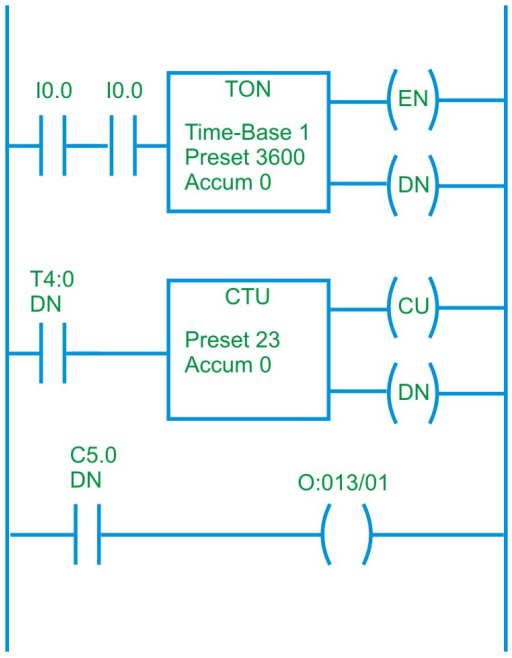

Figura 5.30: Timer + Contador

5.4.5 Sequenciador

Um seqüenciador é um dispositivo utilizado para dar um seqüência de operação a um determinado processo, suponha nós queremos que um motor seja ativado depois que tenham passados 5 segundos do momento que o processo foi iniciado, esse motor deve ficar ligado até que cheguemos a 10 segundos, daí um segundo motor é ativado e continuará ativado pelos próximos 10 segundos, mas passados 5 segundos da ativação do segundo motor, um terceiro motor deve ser ativado e também ficar ativado durante 10 segundos, após o terceiro motor ser desligado um quarto motor deve ser ligado durante 5 segundos. O diagrama de tempo desse processo é mostrado na figura 5.31.

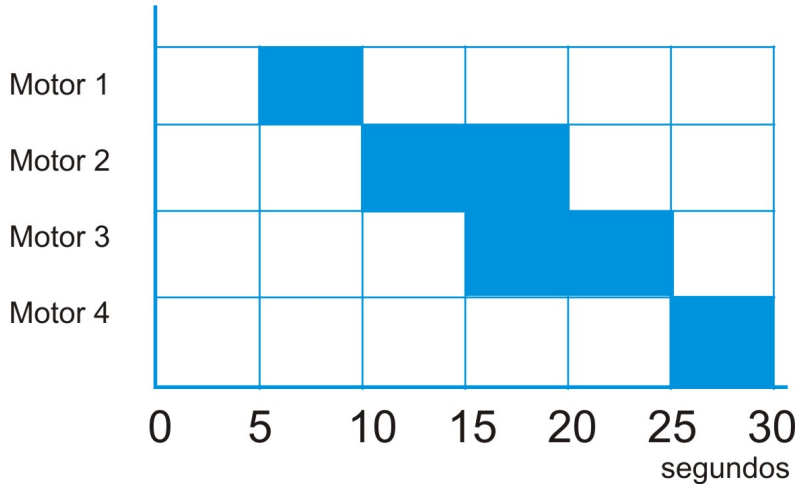

Figura 5.31: Processo utilizando Seqüenciador

Um exemplo de seqüenciador utilizando um CLP da Toshiba é mostrado na figura 5.32. O bloco STIZ é responsável pela seqüência, quando iniciado, R500 é ativado ativando depois o R501, que por sua vez ativa R502 e a saída Y020, o R502 ativa então a saída Y021 e da inicio a um On-Delay-Timer, após passado o tempo necessário R503 é ativado e esse ativa a saída Y023 e também na seqüência ativa o R504 e daí segue a seqüência até o final do programa.

5.5 Registradores de Deslocamento

O termo Shift Register (Registrador de Deslocamento) se refere a agrupamentos de dispositivos internos, tais como Relés Internos, que podem armazenar dados e esses dados podem ser deslocados entre os vários bits desse

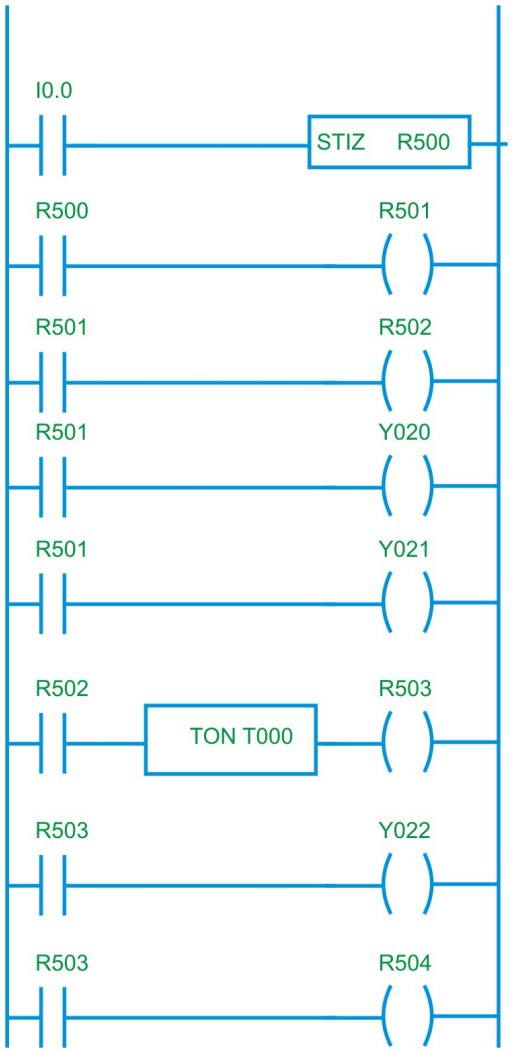

Figura 5.32: Processo utilizando Seqüenciador para CLP-Toshiba

registrador. Um registrador é portanto um grupo de relés internos agrupados, normalmente 8, 16 ou 32, cada relé interno pode se apresentar como aberto ou fechado, 0 e 1 respectivamente, a figura 5.33 mostra um registrador de deslocamento de 8 bits. Cada bit em 1 significa que o relé interno referente aquele bit está fechado, quando 0 está aberto.

Com registradores de deslocamento é possível deslocar dados gravados, esses registradores possuem 3 entradas; Load, para carregar os dados no registrador, Uma entrada de comando para o deslocamento dos bits, e um Reset para apagar os dados gravados no registrador. A figura 5.34 mostra

Figura 5.33: Registrador de Deslocamento

como funcionam os sinais de entrada do registrador de deslocamento.

Figura 5.34: Comportamento de Registrador de Deslocamento

5.5.1 Programando um Registrador de Deslocamento

Considere um registrador de deslocamento de 4 bits, representado em ladder pela figura 5.35, a entrada In3 é usada para resetar o registrador, a entrada In1 é usada para entrar com o valor do primeiro relé interno do registrador, a entrada In2 é usada para deslocar os bits do registrador, cada um dos relés internos é conectado a uma saída, como mostra o programa.

Suponhamos que inicialmente In3 é ativada, dessa forma o registrador é zerado e temos 0,0,0,0, nesse momento ativamos In1 e então o primeiro relé é mudado para 1, resultando dessa forma na seguinte configuração 1,0,0,0, e a saída Out1 é então ativada, note que se In2 for ativado temos 0,1,0,0, e então a saída Out2 passa a ser ativada e assim por diante.

Um programa com registradores de deslocamento para CLPs da Toshiba é mostrado na figura 5.36. O Toshiba R016 é o endereço do primeiro relé do registrador, o (08) indica o tamanho do registrador, D é usado para Data Input, ou seja carregar dados no registrador, S é usado para deslocamento, E para habilitar ou resetar as entradas e Q para saída de dados.

A figura 5.37a mostra o símbolo padrão IEC 1131-3 para registradores de deslocamento, O valor a ser deslocado esta na entrada IN e o tamanho do deslocamento é apresentado na entrada N, a figura 5.37b mostra o símbolo

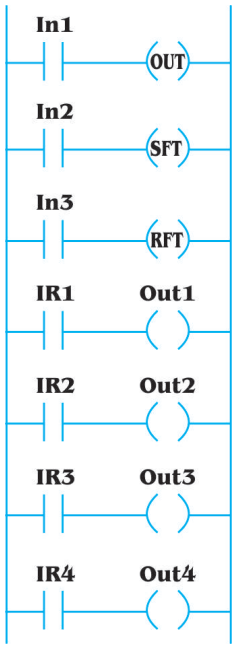

Figura 5.35: Comportamento de Registrador de Deslocamento

para registradores de deslocamento utilizado pela Siemens. O deslocamento pode ser realizado tanto para direita como para esquerda.

5.6 Manipulação de Dados

Até esse momento foram explicados vários dispositivos internos dos CLPs, tais como Timers, Relés Internos e Contadores, todos esses dispositivos trabalham diretamente com bits de dados, agora explicaremos como se da a manipulação de blocos de dados, chamados de words.

5.6.1 Transferencia de dados

Quando desejamos copiar dados de um local na memória para outro utilizamos funções de transferência, a figura 5.38 mostra como fazer isso em Ladder para CLPs da Siemens, aqui os dados em IN são movidos para o local OUT quando EN está ativado.

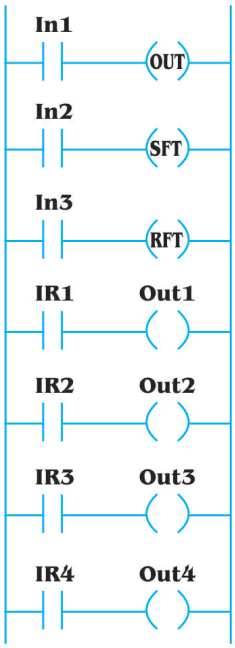

Figura 5.36: Programa com Registrador de Deslocamento/CLP Toshiba

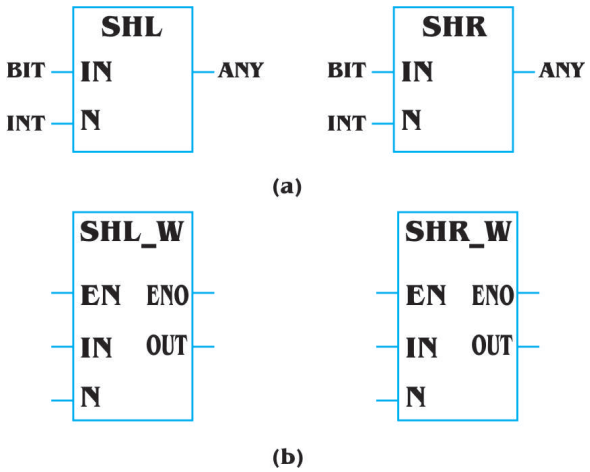

Figura 5.37: (a) Padrão IEC 1121-3, (b) Siemens RDs

5.6.2 Comparação de dados

Em muitas ocasiões desejamos fazer comparação entre dados, sejam esses dados lidos com dados armazenados, comparação entre dados armazenados e dados de processamento, um exemplo de comparação entre dados lidos e ar-

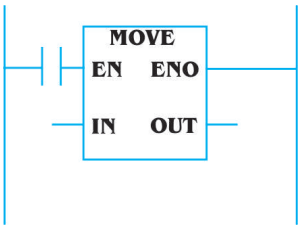

Figura 5.38: Transferência de Dados Siemens

mazenados é o da leitura de temperatura em uma caldeira, no programa podemos colocar um valor fixo que é constantemente comparado com um valor lido por um sensor, após a comparação uma ação é executada de acordo com o resultado da operação. Em CLPs da Siemens utilizamos o bloco mostrado na figura 5.39, nesse caso os valores comparados são as entradas IN1 e IN2 o bloco põe na saída 1 ou 0 dependendo do resultado da comparação.

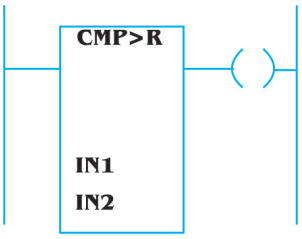

Figura 5.39: Comparação de Dados Siemens

5.6.3 Funções Aritméticas

Dependendo do CLPs podemos encontrar desde apenas funções de soma e subtração até funções um conjunto de todas as funções básicas mais outras funções como exponencial. A figura 5.40 mostra o formato básico de uma função de adição para CLPs da Siemens, a função aritmética é executada quando EN está ativado, esse bloco é o mesmo para funções de subtração, divisão e multiplicação.

5.6.4 Sistemas de controle

Existem duas formas básicas de fazer o controle de um determinado processo, controle em malha aberta e controle em malha fechada, imaginemos o processo de controlar a temperatura de um fluído em uma caldeira.

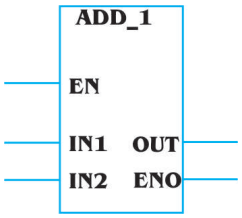

Figura 5.40: Blocos de funções aritméticas

Controle em Malha Aberta

O controle em malha aberta acontece da seguinte forma, um valor de temperatura é selecionado e então o sistema é colocado pra funcionar com uma potência capaz de levar o fluído aquela temperatura desejada, por exemplo 150 graus Celsius, no entanto, qualquer perturbação no processo (por exemplo adição de mais fluído a caldeira) irá perturbar o sistema e a mesma potência não será mais capaz de elevar o fluído a temperatura antes desejada a figura 5.41 mostra o esquema de um controle em malha aberta.

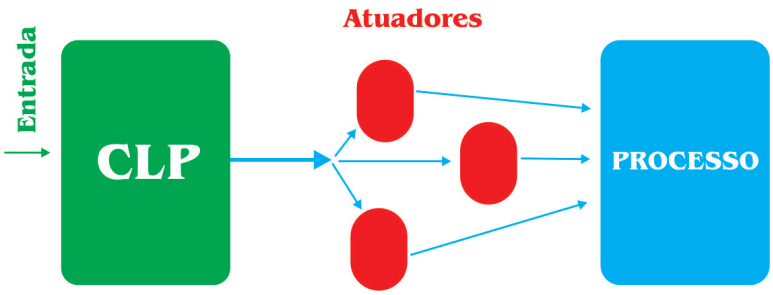

Figura 5.41: Controle em malha aberta

Controle em Malha Fechada

O controle em malha fechada apesar de ser mais complexo, é mais eficiente e mais indicado em sistemas que temos perturbações, como por no exemplo anterior ou a temperatura de uma sala, na qual temos constante entrada e saída de pessoas, dessa forma após ser selecionada uma temperatura, o sistema faz uma leitura da temperatura atual, compara com a temperatura desejada, só então o próprio sistema ajusta a potência na qual o sistema irá funcionar, ou seja dependendo do erro entre temperatura lida e temperatura

desejada o sistema "escolhe"em qual potência irá operar a figura 5.42 mostra um esquema de um controle em malha fechada.

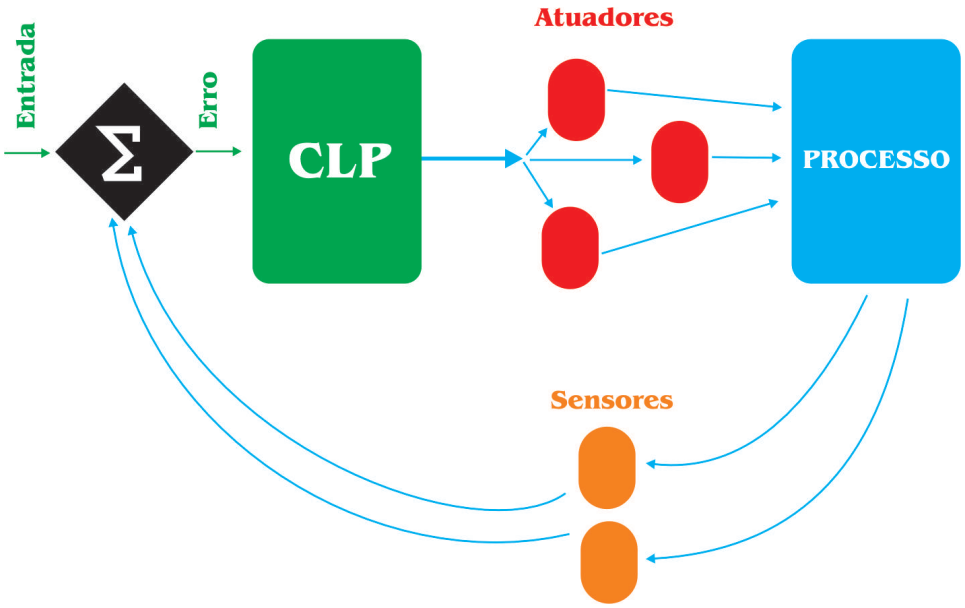

Figura 5.42: Controle em malha fechada

Tipos de Controladores

Dependendo do processo que desejamos controlar e do valor que estamos dispostos a pagar pelo sistema de controle, podemos escolher dentre várias arquiteturas para o sistema de controle, por exemplo controlador proporcional (P), nesse caso temos a seguinte equação para esse tipo de controle.

$$Erro = |ValorLido - ValorDesejado|$$

O erro então é multiplicado por a constante do controlador proporcional.

$$SaidaControlador = KxErro$$

Outro tipo de controlador que pode ser utilizado, é o controlador PID, um pouco mais complexo mas que trás um conjunto muito maior de benefícios, muitos CLPs fornecem os parâmetros para o cálculo de controladores como PID, tudo que é necessário é fornecer os valores desejados.

Referências Bibliográficas

[1] W. Bolton. *Programmable Logic Controllers*. Elsevier Newnes, Amsterdan.

[2] Douglas W. Jones. Control of stepping motors.